

PL-TR-96-2002

## SPACE DEBRIS DETECTION AND ANALYSIS

Robert H. Eather  
Cyril A. Lance  
Quan Vu

Keo Consultants  
27 Irving St.  
Brookline MA 02146

26 February, 1996

Final Report  
08 April, 1992 - 30 October, 1995

DTIC QUALITY INSPECTED 4

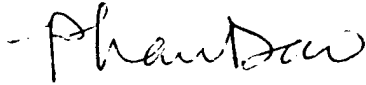
Approved for public release; distribution unlimited



PHILLIPS LABORATORY  
Directorate of Geophysics  
AIR FORCE MATERIEL COMMAND  
HANS COM AIR FORCE BASE, MA 01731-3010

19960813 160

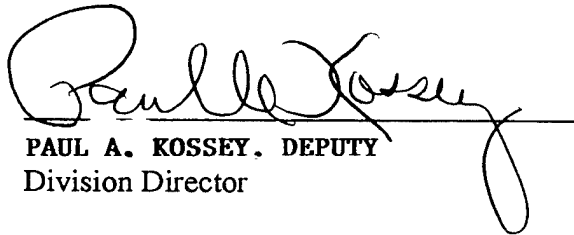
"This technical report has been reviewed and is approved for publication"



PHAN D. DAO  
Contract Manager



DAVID N. ANDERSON  
Branch Chief



PAUL A. KOSSEY, DEPUTY  
Division Director

This report has been reviewed by the ESC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center (DTIC). All others should apply to the National Technical Information Service (NTIS).

If your address has changed, if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/IM, 29 Randolph Road, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 26 February 1996		3. REPORT TYPE AND DATES COVERED FINAL (8 April 1992-30 October 1995)
4. TITLE AND SUBTITLE  Space Debris Detection and Analysis			5. FUNDING NUMBERS PE61102F  PR 2295 TA 24 WU AA	
6. AUTHOR(S) Robert H. Eather Quan Vu Cyril A. Lance			Contract: F19628-92-C-0070	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Keo Consultants 27 Irving St. Brookline MA 02146-7744			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Phillips Laboratory 29 Randolph Rd. Hanscom AFB MA 01731-3010 Contract Monitor: Dr. Phan Dao / GPIM			10. SPONSORING/MONITORING AGENCY REPORT NUMBER PL-TR-96-2002	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  A 40 cm telescope with an intensified CCD detector and video recording was used to detect space-debris objects. Participation in the ODERACS experiment showed that objects as small as 5 cm could be detected with this size telescope. Image analysis techniques were tested to enhance the low S/N debris streaks in the recorded video signals, and the Hough transform was selected as the best technique. A hardware implementation of this transform was carried out using DSP chips, and it was demonstrated that real-time processing should be achievable with parallel processing with five DSP chips. Finally, an analysis was undertaken to use a known debris population distribution to project the population into the future so as to predict debris detection rates from a particular ground observation site. It was concluded that seasonal effects will compromise the usefulness of debris observations from a particular site as an indicator of any systematic changes in the debris population.				
14. SUBJECT TERMS  Space Debris, Space Debris Modeling, Hough Transform			15. NUMBER OF PAGES 82	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	

## Table of Contents

	Page
1. Introduction	1
2. Optical Detection of Space Debris	2
2.1 Optical Detector	2
2.2 Image Analysis Facility	3
2.3 The ODERACS Experiment	7
3. Future Telescope/Detector Development	15
4. Image Analysis of Debris Data	17
5. Hardware Implementation of Image Analysis of Debris Data	19
5.1 System Overview	20
5.2 Hough Transform Simulation Using MATLAB	22
5.3 Hough Transform Implementation on TMS320C40 DSP	24
5.3.1 DSP Selection	24
5.3.2 Software Methodology	26
5.3.3 Results and Analysis	27
(a) Processing Results	27
(b) Timing Results	27
(c) Timing Analysis - DRAM Page Faults	28
(d) Timing Analysis - Processing	29
(e) Summary of Timing Analysis	30
(f) Conclusions	31
5.4 Closed Loop Realization of the ARDD	33
5.4.1 System Setup	33
5.4.2 Synthesized Data	34
5.4.3 Pre-Recorded Data	36
5.4.4 Analysis and Conclusion	40
5.5 Future Development Efforts	42
5.5.1 Approaches	42
5.5.2 TMS320C40	42
5.5.3 TMS320C80	45

## Table of Contents (Ctd.)

	Page
6. Spatial Density Modelling	48
6.1 Objectives	48
6.2 Two-line Element Sets	49
6.3 Software Components Setup	49
6.4 Data Analysis	50
7. Space Debris population Simulation	51
7.1 Overview	51
7.2 Methodology	51
7.3 Simulation of Observations	63
7.4 Look-Angle Calculations	69
7.5 Conclusions	77
8. References	78

## 1. Introduction:

There are currently over 7500 objects circling the earth with radar cross section large enough ( $\sim 10$  cm) to be tracked by the US Space Command space surveillance sensor network and about one half of that could be classified as space debris. A mass plot of all these objects (where the dots representing bits of debris are necessarily enormously exaggerated in size) is shown in Figure 1 (Teledyne Brown Eng., 1986). The number of smaller objects not measured by radar, but still considered hazardous to orbiting assets, is much higher. In the size regime below 0.5 mm, meteors dominate the collision flux (encounters per unit area per unit time) experienced by spacecraft, but for objects larger than 1 mm, collision probability is dominated by orbital debris which is confined to a much smaller volume near earth. Collisions in earth orbit occur at velocities of  $\sim 15$  km/sec, so an object such as an errant bolt could destroy a satellite or endanger astronauts. The greatest density of debris resides in the region 500-1500 km above the earth's surface. To date not a single satellite has been lost owing to space debris, but debris poses a small but growing risk to space activities. Therefore there exists a need to define, monitor and mitigate the debris problem in this size and height regime. Consequently, related research work is being conducted and supported by the DOD.

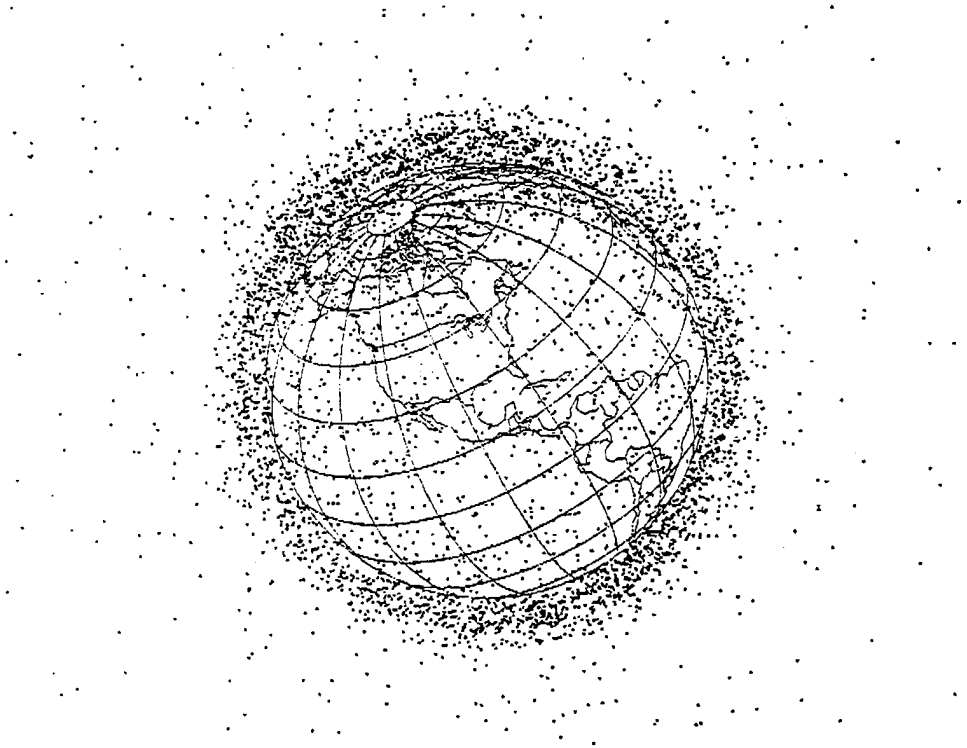


Figure 1: Schematic Representation of Space Debris around the Earth

To properly assess the space environment and model the risk posed by collisions between space assets and man-made space debris, new sources of data have been planned. Three questions to be addressed by these efforts are:

- what is the size of the debris population?
- how accurately do measurements from an individual ground site represent the total debris population?
- how does the debris population impact operational and planned space programs?

The amount of small sized debris obviously needed quantification because it is expected to contribute the most to the total population of space objects in Earth orbits. The Air Force Phillips Laboratory (PL) is undertaking a concerted effort to assess the debris environment by optical measurement and modeling. Debris is measured with optical telescopes operated by the MIT/Lincoln Laboratory in Socorro, New Mexico, and the PL Laser and Imaging Directorate, in Maui, Hawaii. Keo Consultants is assisting PL's effort in the area of debris measurement and image processing.

This report documents Keo's efforts in three areas related to debris assessment:

(i) Optical detection of space debris: A telescope equipped with a CCD or imaging detector can be used to detect space debris. With the telescope fixed or in sidereal tracking mode typically at a high elevation angle, orbital objects under proper solar illumination conditions traversing its field-of-view can be recorded. The proper conditions exist at dawn and dusk when the lower atmosphere is in darkness and the high flying debris objects are still illuminated by the sun. Debris tracks typically exhibit low signal to noise, so sophisticated detectors (image intensified) and sophisticated image analysis techniques are required to identify events above sky and system noise.

(ii) A feasibility study of the use of state-of-the-art microprocessors in automated detection of debris streaks recorded at dawn and dusk with the telescope in a quasi-stationary position. In a typical dawn or dusk session with a large (~ 100 cm) telescope, objects are detected in the field-of-view at the rate between 5 and 20 objects per hour. To date, real-time detection of these debris, to a high level of sensitivity, is still performed by an observer. Immediately after detection, the observer could also track the debris by slewing the telescope in the direction of the "perceived" streak. Tracking refers to the locking of the telescope boresight with the orbiting object based on video information. While tracking is done by electronic means, the critical task of detecting and bringing the object into lock is still performed by an operator. Once tracking is established, an accurate orbital element set can be constructed. That set defines deterministically the orbit of the object and uniquely defines that object. Because detection and telescope slewing is done by a

human operator, it is subject to errors related to lapse of attention and fatigue. Also, note that the observer is required to control the tracking telescope to establish tracking following the detection step. Having a human in the tracking process is a drawback because he is neither fast nor accurate in estimating the detected object's motion and controlling the telescope. This results in a low success rate in tracking of detected objects. Even in non-real time operation, in which tapes recorded can be screened afterward to determine the detection rate and the approximate element sets, an automated system is highly desirable. Human screening is still an expensive procedure and computer processing is preferable whenever possible.

(iii) Analysis of available debris population data as obtained from the NORAD tracking system. The NASA bi-daily two-line element sets were used to create a simulation of the distribution of known (trackable) debris elements in the earth's atmosphere. From this distribution, the goal was to create a histogram of "sightings" for selected locations and fields of view, and to determine if there were any trends over time that deviated from a constant level of detection.

## **2. Optical Detection of Space Debris**

The aim of the space debris detection and recognition project was to optically detect and track space debris of a size that radar is unable to detect ( $<10$  cm). This part of the contract as originally negotiated had two main objectives:

- (i) To build a new optical detector (similar to the intensified CCD camera built by Keo under Contract F19628-91-C-0054, see Eather, 1993) for installation at an astronomical site (e.g. AMOS in Hawaii)
- (ii) To set up a central image analysis facility at Phillips Laboratory to analyze debris video tapes.

### **2.1 Optical Detector:**

In the first Quarter, we evaluated the optical and mechanical specifications for the AMOS telescope in Hawaii, and anticipated no problems in integrating an intensified camera to the telescope. However, Air Force arrangements had not been made to use this telescope at that time. During the third Quarter, it was decided by the Air Force to curtail the observational program, so as to emphasize analysis of existing data sets and extend theoretical analyses. Thus, objective (i) above was not pursued further, and optical measurements were pursued with available equipment from a previous contract.



An image-intensified CCD imager, coupled to a 40 cm telescope with a 2° field-of-view, was available from an earlier Keo contract. Also available was a manually controlled azimuth/elevation mount, with scale readout of the azimuth and electronic inclinometer readout of elevation (see Section 2.3 following).

Site evaluation determinations considered the extreme light pollution in and around the Boston metropolitan area. Due to driving time from Phillips Laboratory and considerations of available space and authorization, we settled on Millstone Hill in Westford, MA as the best site. We were able to place the equipment in a secure building in the middle of a clearing that provided near horizon visibility. The darkness of the sky was poor compared to an astronomical site, and the highest magnitude star that could be detected on the real-time video was  $m \sim 10-11$ . However, the site was acceptable for initial calibration and for tracking practice.

In order to transport the 40 cm telescope and mount to other astronomical sites, we had two foam-lined shipping boxes made that could withstand the rigors of transportation. The telescope mount could be placed in one box in an upright position while the telescope was laid in a horizontal position within the other box. A third shipping box for the electronics rack was available from the earlier contract.

## **2.2 Image Analysis Facility:**

The items purchased to accomplish the video acquisition, storage and subsequent analysis were a Panasonic AG-7750 S-VHS Recorder/Player with AG-F700 Time Code Inserter/Reader, and a Panasonic AG-7355 S-VHS VCR with built-in frame store. Also, to facilitate easy location on the video tape of space debris streak data we purchased an AG-232TC serial interface accessed by the computer serial port. The time code was stamped using a Panasonic AG-F700 time code generator/reader. To complete the acquisition package for data analysis we purchased an overlay framegrabber from Imaging Technology to digitize the video data. To accommodate the large storage needs for digitized images we purchased a Wangdat digital audio tape (DAT) with storage capacities of 2MB uncompressed and 4MB compressed.

We also purchased a computer to perform processing and noise removal on images that were acquired by the 40 cm telescope. The computer chosen was a Gateway 2000 80486 running at the clock doubled speed of 50 megahertz. This fast (as of 1993) personal computer speed was needed to facilitate computationally intensive image processing algorithms such as noise and clutter removal.

To make the video analysis facility user friendly, we had to consider operating environments used by personnel most likely to use the lab for processing purposes. Once the equipment was in place to accept S-VHS tapes from the optical site, applications were written to

provide the capability to control the tape deck from the computer. We wanted to control the digitizing of the images, and provide certain image analysis operations on the digital image. Some time was spent interfacing the computer to the tape deck so as to control command operations (forward, backward, rewind, fast forward), and implementing the capability to direct the tape deck to go to a certain time stamp on the tape. Several applications were written to provide the basic functionality of digitizing, writing captured images to disk, and reading them back again. All these applications took command line arguments for increased flexibility. The digitization application could grab one video frame or a user-specified number of frames. Additionally, the read from disk application could read one or many files at each invocation.

To allow the above hardware components to work together we investigated and purchased certain software applications. Software of varying levels of sophistication and price to support the framegrabber were investigated from many vendors. Some applications provided a WINDOWS' graphical user interface (GUI) with many image processing functions. Other applications were more focused on certain image analysis algorithms. After evaluation it was decided that for our specific needs these applications contained many functions that we would be paying for but weren't needed. Hence, we decided that it would be better to write our own applications using the ITEX image processing library from Imaging Technology.

Test data that had been acquired in the field was used to develop processing techniques. The S-VHS tape was placed in the AG-7355 and advanced to a location on the tape (noted during on-site acquisition) where a debris object had crossed the field-of-view. Using an astronomy program called "The-Sky", we located the star field corresponding to the frames containing the debris object. Evaluations were made as to the debris object's altitude and azimuth readings by comparing to the background star field. The tape data were digitized on a frame-by-frame basis with each frame stored in a separate file. For a typical debris object this procedure would generate about 90-200 files of one-fourth megabyte each. The digitized data files were first stored to the DAT before any processing occurred. Refer to the following data acquisition procedure diagram (Figure 2) for associated components.

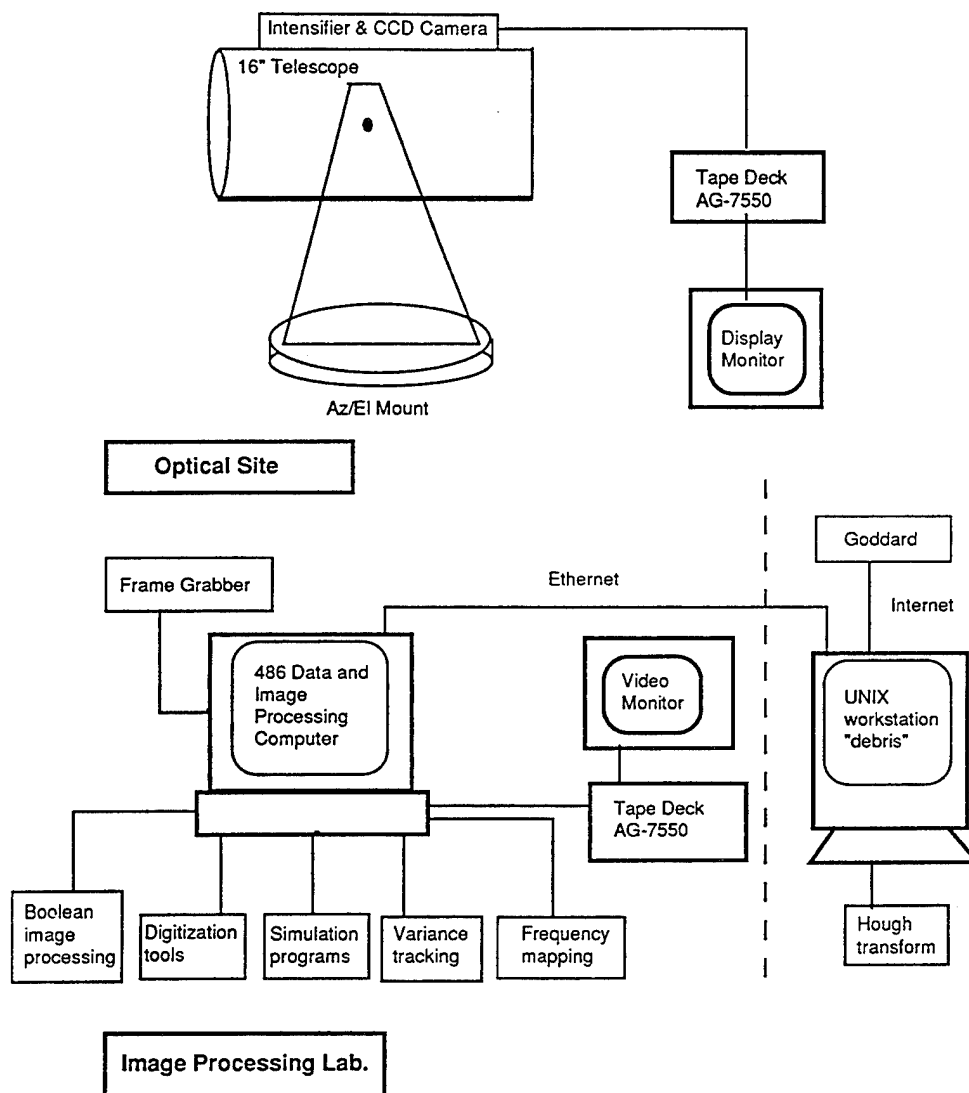
After digitization, the data files were in a format used by Imaging Technology. The format consisted of a header with various fields dedicated to image parameters. The remainder of the file consisted of the image data.

We also needed the ability to process our data on our 80486 or the project SPARC station, code-named 'debris', which contained certain signal processing applications. A program was written to give us the capability to use a program called Interactive Data Language (IDL) that resided on 'debris'. We wanted to use IDL to fill any gaps in desired image processing algorithms that would take too long to develop on our own. Consequently, to use IDL we needed the capability to transfer files over the network. Additionally, the data files had to be translated to the

tag image file format (TIFF) in order to be used on “debris”. The first prerequisite was provided by the FTP protocol while the second requirement was provided by a program we wrote called “image2tif.c”. This program translated image files from Imaging Technologies’ image format to the TIFF format.

An outline of the debris data collection and analysis procedure is shown in Figure 2.

This work on the development of an image analysis facility at Hanscom proceeded early in the contract, but was then put on hold because of the curtailing of the observational program.



**Figure 2: Data Collection and Analysis Procedure**

### 2.3 The ODERACS Experiment:

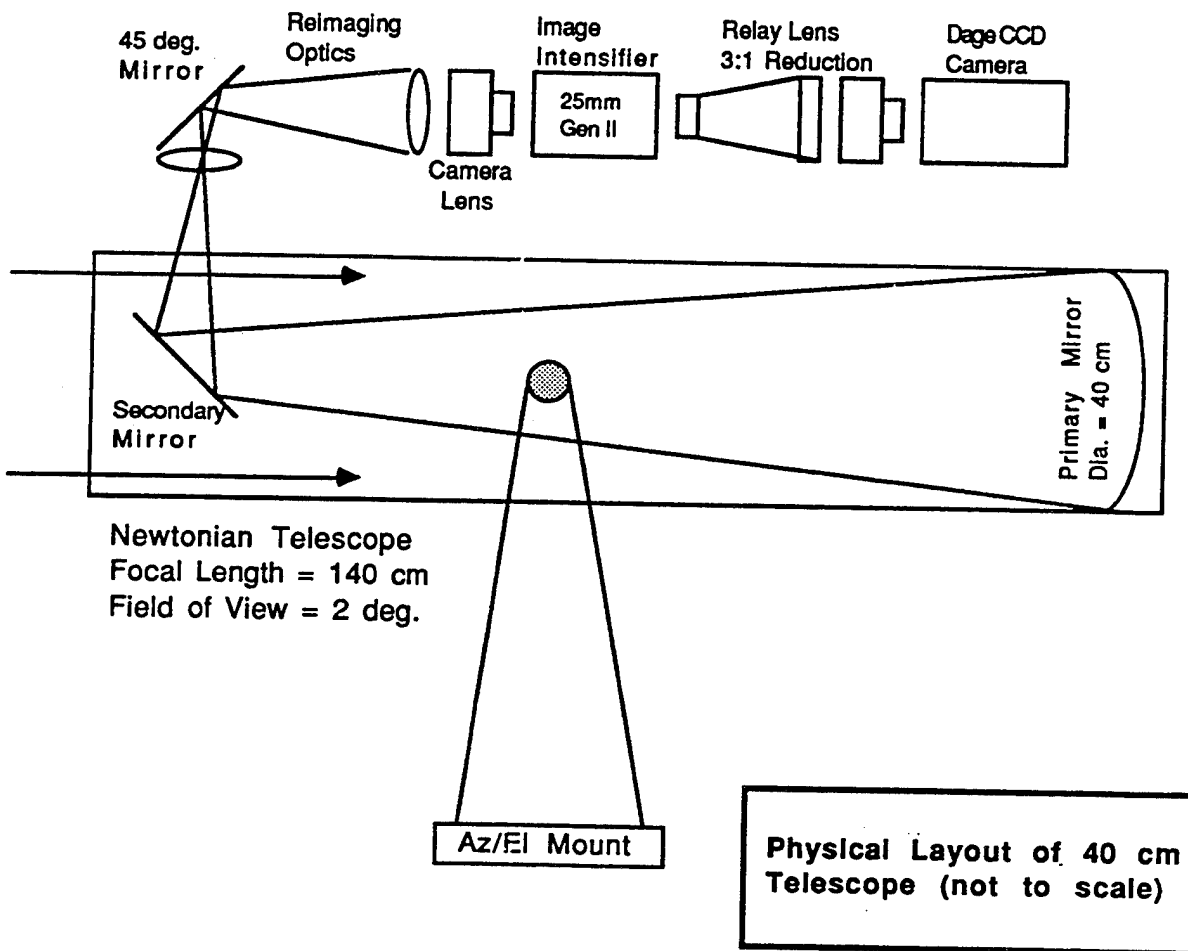
A 40 cm Newtonian telescope with an image-intensified CCD camera, (mounted on a manually operated azimuth-elevation mount) was constructed under an earlier contract, and used for debris observations from Haystack Observatory near Boston (Eather, 1993). A schematic of the telescope and video system is shown in Figure 3, 4 and 5, and system parameters are summarized in Table 1. This telescope was installed at the Rattlesnake Observatory (operated by Battelle's NW Labs. in Richland, Washington) for the ODERACS experiment. The better seeing conditions at this site (compared to Millstone Hill, MA) allowed stars of magnitude  $m \sim 12-13$  to be detected in the real-time video. This compares with  $m = 15$  under "favorable circumstances" for 100 cm GEODSS telescope operated by the USAF Space Command at Maui, Hawaii (Henize et. al., 1993).

ODERACS involved the deployment of various sized reflecting spheres (5 cm, 10 cm and 15 cm) from the Space Shuttle, so as to provide test targets to evaluate the performance of ground-based optical telescopes and radars in detection of small objects in space. Pointing calibrations were made by sighting on known stars and comparing their altitude and azimuth, given by "The-Sky" astronomical program, with that of the altitude and azimuth readings given by the telescope mount. The mount was aligned so that it could be set in azimuth and elevation with an accuracy of about 0.25 degrees.

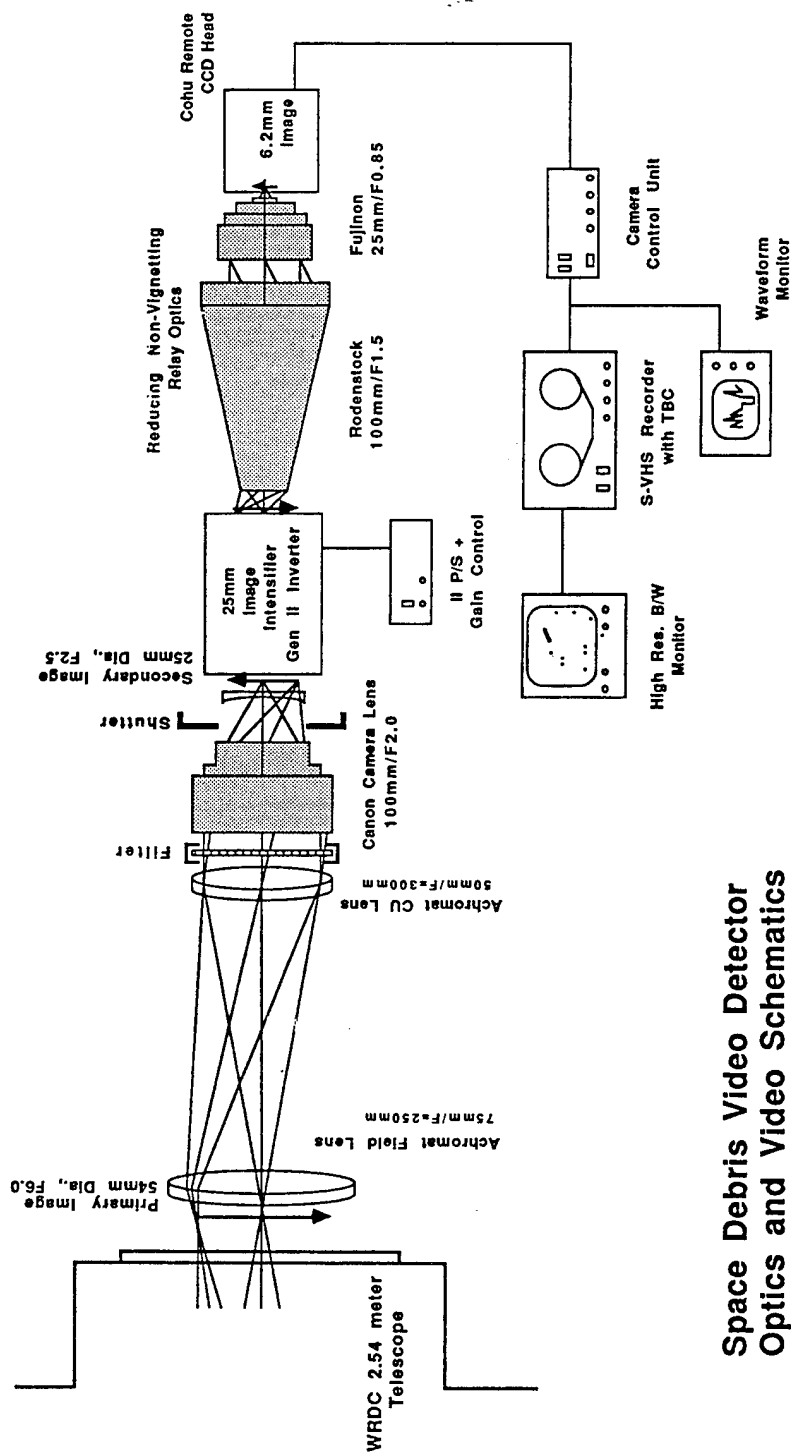
We had previously calculated the expected minimum detectable debris diameter under ideal viewing conditions for a 40 cm telescope (Eather, 1993) and those results are shown in Figure 6 for a S/N in the video signal of 2:1. It may be seen that the minimum detectable diameter was expected to be  $\sim 4-5$  cm.

The spheres failed to deploy from the Space Shuttle in December, 1992 due to battery failure of the container lid mechanism. The telescope was then stored at Rattlesnake in anticipation of ODERACS rescheduling in February 1994.

Successful ODERACS observations were carried out from Rattlesnake Mountain during the period March 9-12, 1994. Various software development was required to support these observations:



**Figure 3: Physical Layout of 40 cm Telescope System**



**Space Debris Video Detector  
Optics and Video Schematics**

**Figure 4: Telescope Optics and Video Schematics**

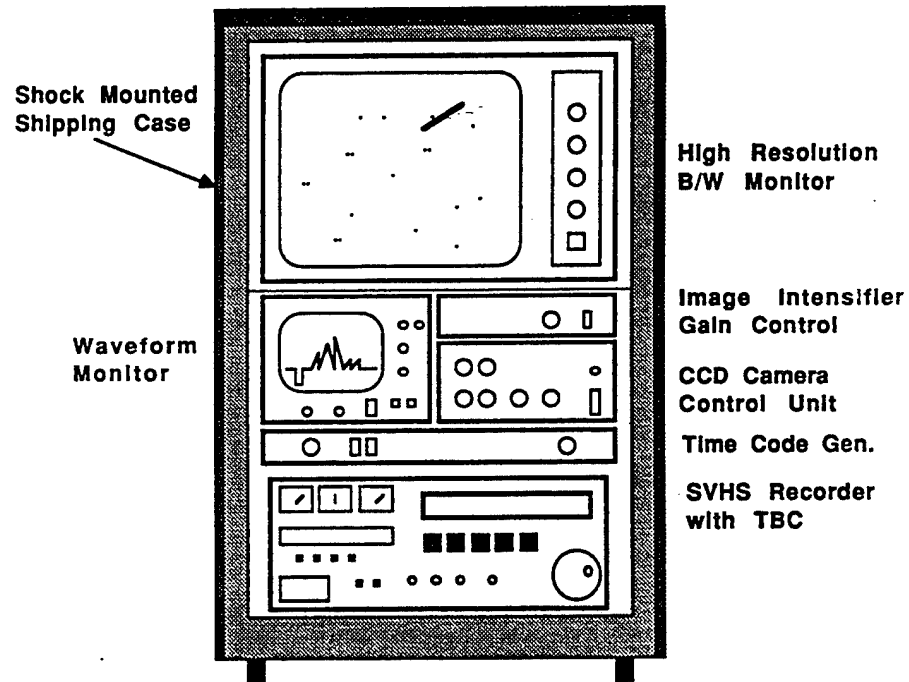


Figure 5: Electronic Rack Layout

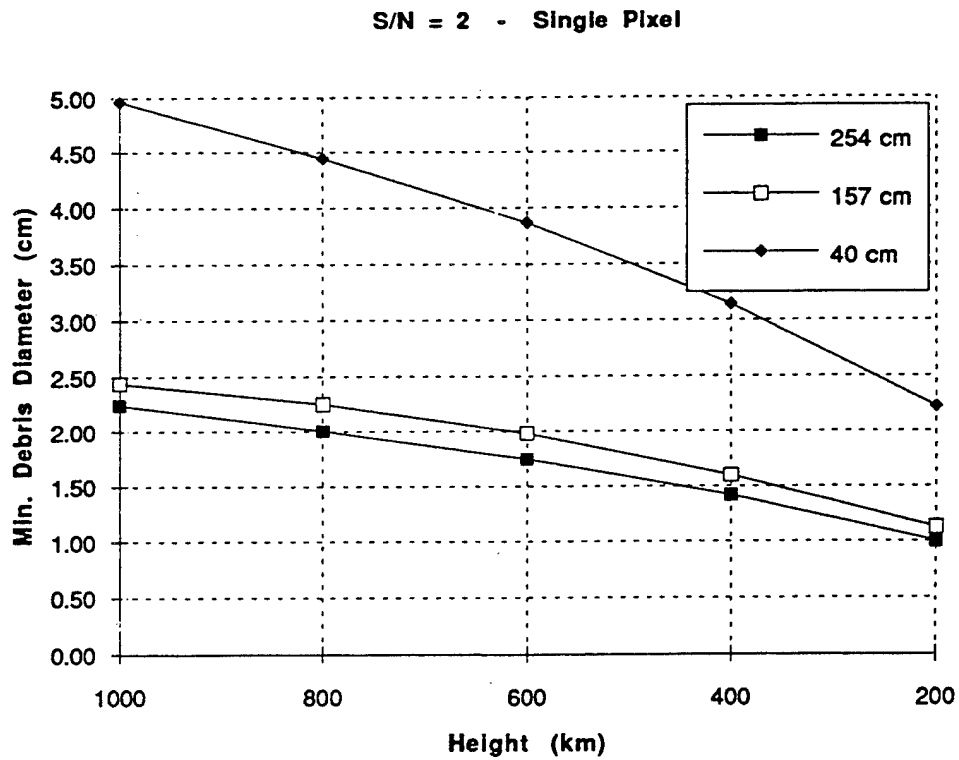


Figure 6: Calculated Minimum Detectable Debris Size for various size Telescopes (Eather, 1993)

**Table 1 - Space Debris Video Detector  
System Parameters**

<b>1. Telescope:</b>	<b>Diameter</b>	<b>40 cm</b>
	<b>Focal Length</b>	<b>140 cm</b>
	<b>F Number</b>	<b>F3.5</b>
	<b>Field of View</b>	<b>2.0 deg.</b>
	<b>Primary Image Size:</b>	<b>50 mm</b>
<b>2. Reimaging:</b>	<b>Achromat Field Lens</b>	<b>f = 250 mm</b>
	<b>Achromat Close-up Lens</b>	<b>f = 300 mm</b>
	<b>Canon Camera Lens</b>	<b>f = 100 mm, F2.0</b>
	<b>Field Curvature Correction</b>	<b>f = -100 mm</b>
	<b>Image Size at Intensifier:</b>	<b>24.0 mm (F2.0)</b>
<b>3. Intensifier:</b>	<b>25mm Gen II Inverted Type</b>	
	<b>Gain</b>	<b>55,000 (2854 source)</b>
	<b>Visible Gain</b>	<b>20,000</b>
	<b>Resolution</b>	<b>36+ lp/mm</b>
	<b>Photocathode</b>	<b>S20R</b>
	<b>Phosphor</b>	<b>P20 (10% faltime = 1 msec)</b>
<b>4. Relay Lens:</b>	<b>Non-vignetting lens combination</b>	
	<b>Rodenstock 100mm/F1.5 +</b>	
	<b>Nikon 35mm/F1.4</b>	
	<b>Image Size at CCD:</b>	<b>8.4 mm</b>
<b>5. Camera: Dage MTI Type</b>	<b>VE-1000</b>	
	<b>Sensor Type Interline Transfer with Microlenses</b>	
	<b>2/3" format, 8.8 x 6.6 mm</b>	
	<b>Pixels 768 (H) x 493 (V)</b>	
	<b>Resolution H</b>	<b>570 tv lines</b>
	<b>V</b>	<b>350 tv lines</b>
	<b>Sensitivity</b>	<b>Full video 0.02 lux (0db gain)</b>
	<b>(Faceplate Illum.)</b>	
<b>6. Recording:</b>	<b>S/N</b>	<b>50db</b>
	<b>S-VHS Format</b>	
	<b>Resolution</b>	<b>400+ tv lines</b>
	<b>S/N</b>	<b>46+db</b>



(i) The astronomical software package "Voyager" was used on a Mac laptop to generate star maps to verify telescope pointing.

(ii) The orbital element database was sourced daily from the INTERNET via modem from NASA, where the format is NASA 2-line excerpted from the shareware SatTrak documentation package.

(iii) These data sets had to be run through orbital tracking software, and three packages were evaluated.

(a) the IBM-PC based SATRAK, which is identical with the NASA package, and we confirmed both packages gave the same results.

(b) a shareware package for the MacIntosh called SATRAK, which was found not to be accurate enough

(c) Another MacIntosh shareware package called OrbiTrack, which was found to be accurate and very useful for interfacing to the Voyager software.

Voyager satellite pass overlays were generated with the OrbiTrack application. OrbiTrack was also used to source random satellite passes for practice observations. However, the OrbiTrack results did deviate slightly from the NASA predictions, so during ODERACS observation periods the IBM version of SATRAK was used.

In most cases we were able to detect the 10 cm and 15 cm spheres, depending on their range and elevation. These spheres were detected by pointing the telescope according to orbital data provided by NASA. However, NASA would not make available parameters for the 5 cm spheres, as this information was classified. We determined that both German and Russian radar facilities claimed to be tracking the 5 cm spheres, and these facilities made the orbital parameters freely available. However, parameters from the two sources did not agree, and we were not successful in detecting the 5 cm spheres. Perhaps we did not have the correct look angles, but the S/N from the 10 cm and 15 cm spheres indicated that there was only a minimal possibility of detecting the 5 cm spheres under ideal viewing conditions and close range passes. This conclusion also agreed with our earlier theoretical predictions (see Figure 6 above) that showed the limit of detectability with a 40 cm telescope would be ~4-5 cm.

The ODERACS results are summarized in Table 2 below:

**Table 2 - Summary of ODERACS Observations**  
**Rattlesnake Mountain, WA [119°35'42"W, 46°23'42"N, Elev = 1088m]**  
**March 9 - 12, 1994**

**Note:** Earliest possible observing time during this window (sky background low enough to turn on intensifier) was about 18.45 LT

Sphere #	Description
22990	6" Chrome Polished
22991	6" Alum Diffuse
22992	2" SS Polished
22993	2" SS Diffuse
22995	4" Chrome Polished
22996	4" Alum Diffuse

**March 9:** Mostly cloudy, but looked through breaks in clouds for known satellites to prove pointing accuracy. We recorded transits of the following satellites:

Time	Az	El	Range	Satellite
1920.00	220.4	15.6	1020	DMSP B5D2-3
1937.00	205.5	77.6	338	Bremstat

Note: Time recorded on tape is 2 hours slow.

**March 10:** Clear, but high winds causing significant telescope jitter. We looked for all six Oderacs spheres, with the following results:

[Sun Angle is the angle defined by the observing site, the sphere, and the sun.]

Sphere #	Time LT	Az Deg.	El Deg.	Range km	Phase Angle Deg.	Norm. Signal	Expected Signal (Norm.)	Detect ?
22990	18:40:13	133.6	33.7	575	47.7	0.160	41.5	No*
22991	18:52:26	135.8	46.6	455	49.5	0.185	76.6	Yes
22992	19:22:28	321.5	58.5	350	77.4	0.165	12.8	No
22993	19:22:52	321.5	77.7	351	75.5	0.115	8.9	No
22994	19:26:47	322.5	70.5	363	82.4	0.150	43.3	Yes
22995	19:29:09	322.7	66.5	373	85.4	0.115	31.5	Yes

\* Not dark

**Table 2 Ctd. - Summary of ODERACS Observations**

**March 11:** Clear, no winds. We looked for all 6 spheres, with the following results:

Sphere #	Time LT	Az Deg.	El Deg.	Range km	Phase Angle Deg.	Norm. Signal	Expected Signal (Norm.)	Detect ?
22990	19:01:28	321.8	74.4	350	95.7	0.143	100.0	Yes
22991	19:14:11	324.3	54.2	413	105.4	0.080	40.2	Yes
22992	19:46:05	330.6	29.3	655	111.4	0.110	2.4	No*
22993	19:46:26	330.7	29.1	658	109.5	0.080	1.8	No*
22994	19:59:34	331.5	27.4	690	111.4	0.150	12.0	Yes*
22995	19:53:00	332.0	26.4	709	112.1	0.080	6.1	No*

\* Not recorded due to operator error in starting recorder

**Calibration Star Field:**

Centered on star SAO:79436 (m = 9.3)      Time: 21:58

RA	=	7hr30.1m	Az	=	240°51'
Dec	=	29°35'	El	=	63°05'

Time	Image Intensifier Gain
2157.00	2
2157.40	3
2158.20	4
2159.00	Off

Then recorded 1 minute of Image Intensifier noise (Gain = 4) (capped telescope) - there is essentially no Image Intensifier noise, only an occasional ion scintillation. All the noise background on the data tapes is real sky background.

**March 12:** Clear, no winds. Strong star scintillation indication poorer viewing. We looked for five spheres, and Bremstat, with the following results:

Sphere #	Time LT	Az Deg.	El Deg.	Range km	Phase Angle Deg.	Norm. Signal	Expected Signal (Norm.)	Detect ?
Bremstat	19:16:45	328.7	34.1	585	113.6			Yes
22990	19:22:49	330.6	29.0	653	123.8	0.135	27.1	No
22991	19:36:01	333.2	23.9	753	124.0	0.055	8.3	No
22994	20:13:50	339.9	15.6	1020				No
22992	20:09:30	333.6	16.3	993				No
22993	20:09:43	333.6	16.3	994				No

**Table 2 Ctd. - Summary of ODERACS Observations**

**Calibration Star Field:**

Centered on star SAO:79436 (m = 9.3)      Time: 20:32

RA	=	7hr30.1m	Az	=	155°19'
Dec	=	29°35'	El	=	71°54'

Time	Image Intensifier Gain
2030.00	1
2030.40	2
2031.20	3
2032.00	4
2032.40	Off (Video Noise)
2034.00	4 - Telescope capped
2035.00	Off

### 3. Future Telescope/Detector Development:

To detect smaller objects than presently possible with the 40 cm telescope and intensified CCD camera, the following improvements could be considered:

3.1 Telescope Size: For a point source, the signal is directly proportional to the telescope aperture i.e. to its diameter  $D^2$ . It would be practical (though considerably more expensive) to upgrade to say a 100 cm telescope, increasing sensitivity by a factor of  $(100/40)^2$  or  $\sim 2$  stellar magnitudes. (As mentioned in Section 1, it was intended to build a detector to attach to the 100 cm AMOS telescope in Hawaii, but this part of the Contract was canceled by the Air Force.)

3.2 Image Intensifier Cathode: The only other way to increase sensitivity would be to increase image intensifier cathode quantum efficiency. Presently, a Gen II type intensifier is used, with an extended S-20 (S20R) cathode, whose quantum efficiency is shown in Figure 7. Gen III GaAs photocathodes are now widely available, and have increased quantum efficiency above  $\sim 500$  nm by a factor of 2-3. Although these cathodes exhibit higher thermal noise, cooling would not be necessary for real-time video, as sky background would still be the dominant source of noise. Extended blue versions of this cathode are currently becoming available, so it is possible that the effective quantum efficiency, folded into the reflected solar spectrum, could be increased by a factor of 3, or more than 1 stellar magnitude.

3.3 Conclusion: If both improvements described above were to be implemented, it should be possible to detect stars down to  $m \sim 16$  on real-time video. Based on our earlier calculations (Eather, 1993 and see Figure 6 above), this should allow debris objects as small as  $\sim 1$  cm to be detected from favorable astronomical sights under no-moon viewing conditions.

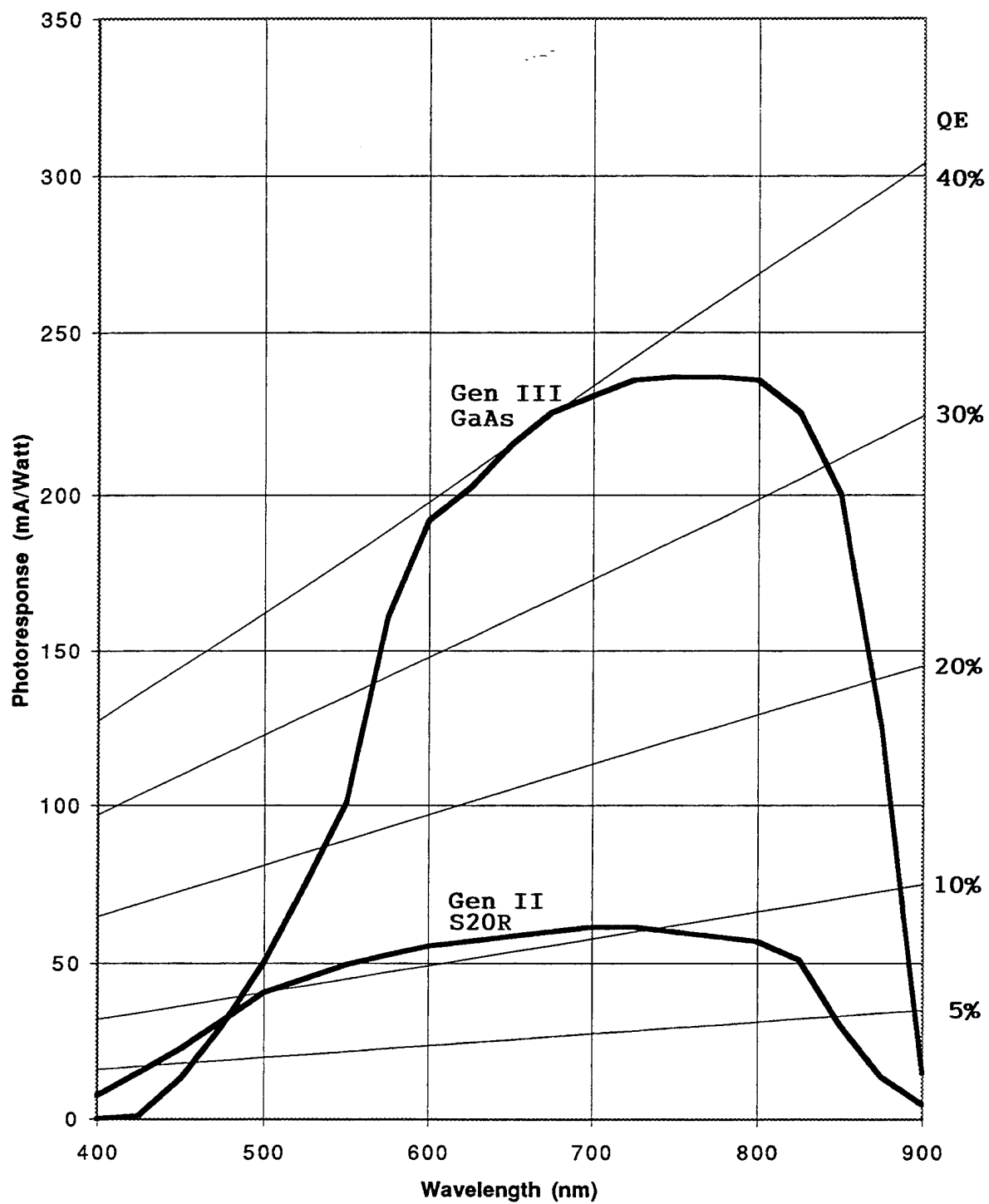


Figure 7: Quantum Efficiency of Gen II (S20R) and Gen III (GaAs - extended Blue) Intensifier Cathodes

#### 4. Image Analysis of Debris Data:

Because of its high orbital velocity, an orbiting object such as space debris leaves a streak in the video image. Its signature is a line of bright picture elements (pixels) hence referred to as a streak. In principle, the streak is dimmer than the equivalent stationary object making detection more difficult and less sensitive. A technique using multiple pixels can be developed to increase detection sensitivity in a low signal to noise ratio (SNR) situation.

A first task in 1992 was to familiarize ourselves with the current state-of-the-art in streak detection. Some of the research was accomplished at the Geophysics Laboratory Library. Internal reports by Lincoln Laboratory as well as familiar books on image processing were used. Also, discussions with personnel from The Analytical Sciences Corporation (TASC) proved to be very useful. The ultimate aim was to identify an appropriate image-processing algorithm that could then be implemented in hardware for real-time image analysis for space-debris streak detection.

We reviewed a Lincoln Laboratory report (Chu, 1989) titled "Detection of Small Moving Objects" describing the SBV algorithm. This algorithm proposed using a "projection method" to reduce the dimensionality of the huge amounts of data involved. This proposal was backed up by analyses that showed these projections were valid for representing the data without loss of relevant information. However, this algorithm involved a computationally intensive two-step process, and we were looking for real-time response implementation. SBV was informative insofar as outlining possible methods with which to attack the massive amounts of data.

Additionally, we would be working in a very low signal-to-noise ratio (S/N) and therefore a thorough understanding of state-of-the-art noise removal techniques was essential. The major thrust in this field has been using median filtering methods because of the "impulse noise" at the detector. Median filters have the attractive property of suppressing impulse noise while preserving edges. The median filter, however, does not preserve thin line details (TLDs) such as space debris streaks. Because of the TLD problem many researchers are investigating hybrid median filters such as: combination median filters (CM), multilevel median filters (MLM), and single adaptive median filters (SAM). The CM incorporates the MLM and the normal median filter, as well as a directional median filter and proved to be of possible use. The main idea of a median filter is in taking a block of pixels and setting each of those pixels in the block to the value of the median of the group. The statistical median is much less sensitive than a statistical average would be because an extreme value isn't going to affect the median pixel's value whereas it would affect a statistical average value. Therein lies its value as a filter to smooth radical outlying values.

##### (a) Algorithm Research:

Research was carried out to determine possible algorithms to use for processing the digital images for streak information. All of the algorithms we adapted or were written in the C/C++

language using the Microsoft C/C++ version 6 & 7 compiler. After these modules were compiled and linked into executables they were included as program item icons under the Windows operating system. These programs were then able to run in a DOS window.

Three algorithms types that were developed or adapted: (i) The frequency method, which is a technique described in various image processing texts. This algorithm was implemented under the statistics package 'MATLAB' as this package has excellent speed for such computationally intensive procedures as a Fourier transform poses for a data set of anything other than small size. (ii) The variance method was developed by Keo software engineers, and (iii) the Hough line transform method was developed by personnel from The Analytical Sciences Corporation (TASC) under a separate contract, and provided to Keo.

(i) Frequency method:

The frequency method was developed to determine if there were any changing components in the scene being analyzed. The method is implemented by summing each column of pixels to reduce the dimensionality from two to one, leaving a one-dimensional array of intensity values. The one-dimensional array is then mapped to a set of exponential functions and transformed into frequency-space by the Fourier transform. This procedure is done on a frame-by-frame basis, allowing for an output of frequencies of changes to the set of frames. If there is a change of intensities (meaning a moving object) in the set of one-dimensional arrays, then that information is described by a frequency histogram with non-zero components. From those frequencies a determination can also be made as to the velocity of the object. However, this method was very susceptible to noise, both from the detector and the clutter background, and was found to be unstable for practical use with real test data.

(ii) Variance method:

Another method, termed the variance method, involves looking at statistics of differenced frames. A large variance value usually indicates where the debris object is located since the clutter is close to the mean of the differenced frame. Hence, the variance graph showed a relatively high peak at the position of the debris. Once in a while a clutter object (star) would move just enough to allow some high peaks at the clutter movement positions in the variance plots. However, these would soon die out and the differenced debris object's variance would again be the maximum value of the vector. Hence, by keeping track of the x,y positions of the maximum value of the variance vectors one can construct a track of the object's motion.

(iii) Hough line transform method:

The Hough line transform method, proposed and implemented by members of The Analytical Sciences Corporation (TASC) was the preferred technique adopted by Phillips Laboratory personnel. The basic idea behind the Hough transform is to map a line's parameters (slope and intercept) to a co-ordinate system of slope-intercept space. Since a debris object creates a 'streak' in the field-of-view (FOV), its line parameters could be mapped to a point in slope-intercept space. As each point of the line has the same slope-intercept parameters these will be mapped to the same point in the slope-intercept space. By thresholding that slope-intercept point one can determine if a 'streak' is present in the FOV. The algorithm was implemented on Sun workstations and shown to work well under high signal-to-noise scenarios.

## **5. Hardware Implementation of Image Analysis of Debris Data:**

Computers and Digital Signal Processors are projected to eventually replace human operators in the demanding task of detecting orbiting objects in video data. Since electronic imaging sensors are used to collect data about orbital objects and debris, machine detection could be developed to alleviate the labor-intensive task of measurement. That would provide a capability of high value to the network of space surveillance sensors.

As described above, signal integration along the streaking direction can be achieved with the Hough Transform algorithm. To date, the Hough Transform has not been implemented in real-time video system because it requires computing resources not available with computers in the moderate price range. However, Digital Signal Processing (DSP) technology provides a good opportunity to develop semi-tailored image processors with high computing speed at an affordable price.

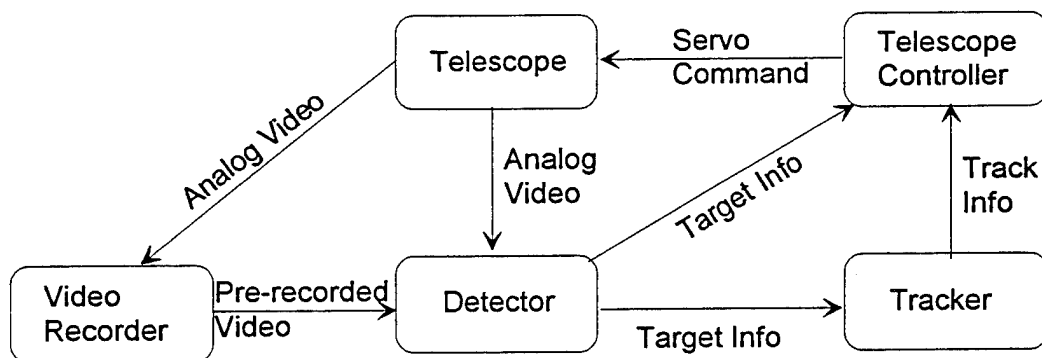
The application of Hough Transform technique on a DSP platform was explored. Engineering realization of the system was undertaken to obtain empirical results complementing the algorithm developed simultaneously at TASC and the Phillips Laboratory (PL). The algorithm was first validated in a MATLAB simulation, producing a set of results meeting expectation and achieving projected sensitivity enhancement. The simulation was then implemented and verified on a single TMS320C40 DSP board. Test results were consistent with the MATLAB simulation. Finally, the Hough Transform was integrated into the system to acquire and process debris data recorded in video tapes.



Results of this sub-scale DSP system operation lead to the conclusion that real time detection can be achieved by using DSP chips chained in a hybrid pipe line architecture. Test results also indicate that a full scale system can be implemented at moderate cost to achieve real-time detection capability.

## 5.1 System Overview

A typical system is depicted in Figure 8. It consists of a single or multiple telescopes, a target detection subsystem, a target tracking subsystem, and a servo controller for the telescope.



**Figure 8: Block Diagram Of Space Debris Detection and Tracking System**

The principles of passive detection are as follows. A telescope equipped with a CCD or imaging detector is used to measure debris. With the telescope fixed or in sidereal tracking mode typically at a high elevation angle, orbital objects under proper solar illumination conditions traversing its field-of-view can be recorded. The proper conditions exist at dawn and dusk when the lower atmosphere is in darkness the high altitude debris is still illuminated by the sun.

In a dawn or dusk session, objects are detected in the field-of-view at the rate between 5 and 20 objects per hour (depending on telescope size). To date, real-time detection of this type of debris to high level of sensitivity is still performed by an observer. Immediately after detection, the observer may attempt to track the debris by slewing the telescope in the direction of the "perceived" streak. Tracking refers to the locking of the telescope boresight with the orbiting object based on video information. While tracking is done by electronic means, the critical task of detecting and bringing the object into lock is still performed by the operator. Once tracking is established, an accurate orbital element set can be constructed. That set defines deterministically the orbit of the object and uniquely defines that object.

Detection and telescope slewing by an operator are subject to inconsistencies and errors related to lapse of attention and fatigue. An operator is not fast and accurate in estimating the detected object's motion or controlling the telescope. This results in a low success rate in tracking of detected objects. In non-real time operation, video tapes recorded can be screened afterward to determine the detection rate and the approximate element sets. Manual screening by an operator is still an expensive procedure and computer screening is preferable whenever possible.

Therefore, an automated system in real-time mode is highly desirable and has the following characteristics:

- Working automatically, without manual processing or data screening.
- Knowing when tracks are instantaneous.
- Automatically measuring object position, velocity and direction.
- Processing in real-time 512x512 format and scaling to 1Kx1K image.
- Detecting discontinuous object tracks.
- Detecting and tracking multiple objects simultaneously.

The key abstraction to automate the process is the orbiting object, because information about an object, once detected, can be fed to other subsystems to provide a completely automatic system. For example, given the positions of an object obtained in several consecutive frames, the object track can be established and its velocity can be derived. Consequently, based on the velocity and current position, the future position of the object can be predicted and used in steering the tracking telescope. Furthermore, intensities or signal levels of the detections can be used to adjust the thresholds automatically and yield a better detection rate. The probability of false alarm is reduced using certain criteria, such as MxN averaging, to ignore or drop the blips.

The starting point to achieve a high degree of automation in real-time operation is to replace the visual detector with an automatic real-time detector as shown in Figure 9. The front-end of the automatic real-time detection subsystem, referred as ARDD, is an analog video signal. This signal is provided by either live video from a telescope or previously recorded signal from a video recorder. The analog video signal is acquired and digitized into digital images of 512x512 8-bit gray scale pixels. The digitized images are then processed by several digital signal processors, implementing Hough transformation, to detect target candidates and provide candidate information to other subsystems.



**Figure 9: Signal/Data Flow of an Automatic Real-time Debris Detector**

The acquiring rate of 30 frames per second is required to match the speed of orbiting debris traveling at approximately 7 km/sec. For an altitude of 300 km, that speed is translated into an angular speed of 1.6 deg/sec for a zenith-pointed telescope. In a 2-degree field of view, mapped to a 512 pixel image, the object travels 410 pixels in a second, or ~12 pixels in a tv frame time. To measure its angular velocity (hands-off information), the length of the streak is needed. Length is measurable when a linear streak begins and ends in the field of view (FOV). That condition is met when the streak integration time is much smaller than  $512/410 = 1.2$  seconds.

## 5.2 Hough Transform Simulation Using MATLAB

The technique used in the Hough transform based on pre-calculated tables called XCOS and YSIN (a common approach used to reduce the number of multiplications and eliminate transcendental operations in the inner loop). Figure 10 is the listing of a MATLAB function including test data that was used to ensure the correct implementation of the algorithm.

```

% _____ HOUGH.M _____
%
% THIS FUNCTION IS USED TO ILLUSTRATE THE HOUGH TRANSFORM TO BE
% IMPLEMENTED IN THE ARDD TESTBED.
%
% USAGE: result = hough()
%         (no parameters are needed - test data is
%         generated within this file).
% FUNCTION CALLS: size,hist,zeros,sin,cos
% _____

%
%function y = hough()
% --- Initialization ---
% -45 degree line:
%pts = [ 9 1 ; 8 2 ; 7 3 ; 6 4 ; 5 5 ; 4 6 ; 3 7 ; 2 8 ; 1 9 ]; % sample input (ordered
pairs)
% +45 degree line
%pts = [ 9 9 ; 8 8 ; 7 7 ; 6 6 ; 5 5 ; 4 4 ; 3 3 ; 2 2 ; 1 1 ]; % sample input (ordered
pairs)
% (2) +45 degree lines
pts = [ 9 9 ; 8 8 ; 7 7 ; 6 6 ; 8 5 ; 7 4 ; 6 3 ; 5 2 ; 4 1 ]; % sample input (ordered
pairs)
pts = pts.*10;
angles = [-20 : 20] ./ 20 .* pi ./ 2 ; % 40 angles from -pi/2 to pi/2
coords = [1 : 128] ; % assume 128x128 image
xcosa = round( coords' * cos(angles) ); % xcosa table
ysina = round( coords' * sin(angles) ); % ysina table

% --- calculations ---

% Max output rho is at angle=45 degrees and X=Xmax, Y=Ymax
% which will be covered by 2 * larger of Xmax and Ymax
% Max negative rho is angle=-90 degrees and Y=Ymax
% which will be covered by Ymax additional bins
% so create an output histogram image with 3 * Ymax
% bins (Xmax=Ymax assuming a square image)
Ymax = size(coords,2);
pointx = pts(ih,1); % extract the next point's X coord.
px = xcosa(pointx,:); % and then it's X*cos(angle) vector

pointy = pts(ih,2); % extract the next point's Y coord.
py = ysina(pointy,:); % and then it's Y*sin(angle) vector

% FOR each angle
for ia = 1:1:size(angles,2)

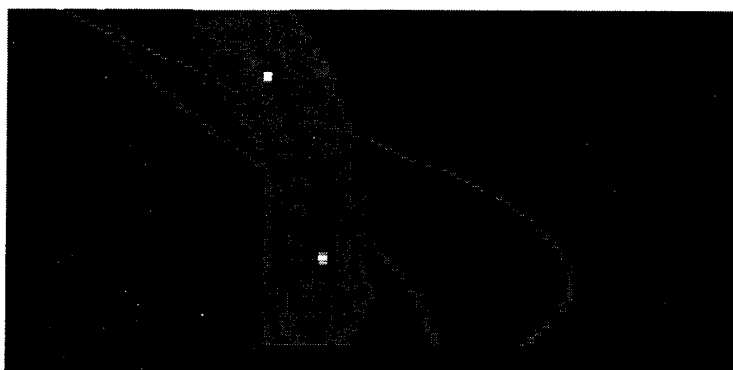
    rho = px(ia) - py(ia) + Ymax ; % calculate RHO for (x,y) at angle(ia)
    output(ia,rho)=output(ia,rho)+1 ; % use RHO to update the histogram output

end
end

```

Figure 10: MATLAB Function of Hough Transform

A simple test case was run under MATLAB environment with 22 ordered pairs as input to the Hough Transform, with a resolution of 90 angles (over 180 degrees). The points were set up as two 10-point lines, at +45 degrees and -45 degrees, and two outlier points. This simulation produced two histogram peaks at the correct locations in the output. The output is shown in a 4 gray-level output [scaled to make the peaks visible] in Figure 11.



**Figure 11: Hough Transform Output**

### **5.3 Hough Transform Implementation on TMS320C40 DSP**

#### **5.3.1 DSP Selection**

Several families of digital signal processors -- INMOS T9000 Transputer, Intel i860, and Texas Instruments TMS320 -- were considered for the system platform. The Texas Instruments TMS320, particularly its C40 generation, was selected because of its inter-processor communication capability, scalability, and wide-support from third party vendors.

The key features of the Texas Instruments TMS32C40 device are:

- Six communication ports for high-speed inter processor communication.
- Six-channel DMA coprocessor for concurrent I/O and CPU operation, thereby maximizing sustained CPU performance by alleviating the CPU of burdensome I/O.
- High performance DSP CPU capable of 275 MOPS and 320 Mbytes/sec. CPU key features include: eleven operations per cycle throughput, resulting in massive computing parallelism and sustained CPU performance, 40-ns or 50-ns instruction cycle times, and 40/32-bit single-cycle floating-point/integer multiplier for high performance.
- Two identical external data and address buses supporting shared memory systems at high data rate, single-cycle transfer.
- On-chip analysis module supporting efficient, state-of-the-art parallel-processing debug.

Inter-processor communication plays an important role in parallel processing. Besides a comprehensive memory interface, the TMS320C40 incorporates on-chip hardware to facilitate high-speed inter processor communications and concurrent I/O without degrading processor performance. This capability is carried out by the six (6) communication ports, each providing 20-Mbytes, bi-directional data transfer operations.

With a direct (glueless) processor-to-processor communication scheme, the C40 is a prime processor for system scalability. Future throughput requirements can be accommodated by connecting additional processors to the system via the communication scheme. This capability has been shown in the market in which one single board can accommodate configuration from a single processor to eight processors.

With strong commitment from Texas Instruments and with more than fifty (50) third party hardware manufacturers, the TMS320 family is indeed widely supported. Hardware components can be purchased directly from the manufacturer as COTS to minimize the cost and risk of hardware fabrication and system integration effort.

In addition, the next generation of TMS320 delivers a computation throughput of 10 to 50 times faster and increases communication bandwidth to several times. The selection of TMS320 family provides a smooth path to upgrade the system from current generation to the next one in order to accommodate the future requirement of processing 2048x2048-pixel images.

### 5.3.2 Software Methodology

A 'C' language description of the Hough Transform as it was to be implemented was provided in an appendix of the original ARDD demonstration. In those pages, a technique was used based on pre-calculated tables called XCOS and YSIN. This technique was first validated in a MATLAB simulation, producing a set of test vectors and associated expected results. This simulation was then translated into 'C' code for the MegaImager board and validated with the same test vectors as were used in the MATLAB simulation. Extensive test support code was added to allow for precise timing experiments, output analysis and result displays.

The Hough transform was implemented using 'C' because it was verified that the optimization that the Texas Instruments compiler provides can achieve timing results very close (within approximately 20%) to that of hand-tuned assembly language. Using a higher level language enabled a more thorough exploration of various algorithm implementation techniques and made integration with vendor's software (MegaImager) much easier.

To perform the timing studies, the on-chip timer of the TMS320C40 DSP was used. This timer uses the internal instruction clock and can be started and stopped by software. To facilitate the use of the timer, two functions were provided with a 'C' callable interface to start and stop the timer, returning the number of timer ticks between the start and stop calls. A timer tick is interpreted to be 2 instruction cycles in duration or 4 oscillator cycles. For a 40 MHz. 'C40 (as used in this demonstration), this translates into a 25 nsec oscillator period and an 100 nsec timer tick period. To generate real-time measurements, therefore, the timer tick count must be multiplied by the 100 nsec timer tick period.

Using the MegaImager demonstration software as a base allowed the majority of the ARDD demonstration effort to be concentrated on the algorithm implementation, rather than the board and PC integration. The MegaImager demonstration software on the PC assumes that output from the 'C40 board will primarily be in the form of image data. In the case of the Hough Transform 2-D histogram output, this is essentially correct and makes it easy to visualize the results [in four level gray scale]. The timing results, however, are scalar values that are not easily translated into image data. To overcome this limitation, the several routines were created to write scalar values as bit-mapped digits into the output image (much like the printf() facilities in the standard I/O 'C' library). Using these subroutines, the timing results are written into the same buffer as the Hough transform output, below the histogram 'image'.

### 5.3.3 Results and Analysis

#### (a) Processing Results

A test case was run with 22 ordered pairs as input to the Hough Transform, with a resolution of 90 angles (over 180 degrees). The points were set up as two ten point lines, at +45 degrees and -45 degrees, and two outlier points. The processing results were verified with the MATLAB simulation and produced two histogram peaks at the correct locations in the output. The output is shown in a 4 gray-level output [scaled to make the histogram peaks visible] in Figure 12.

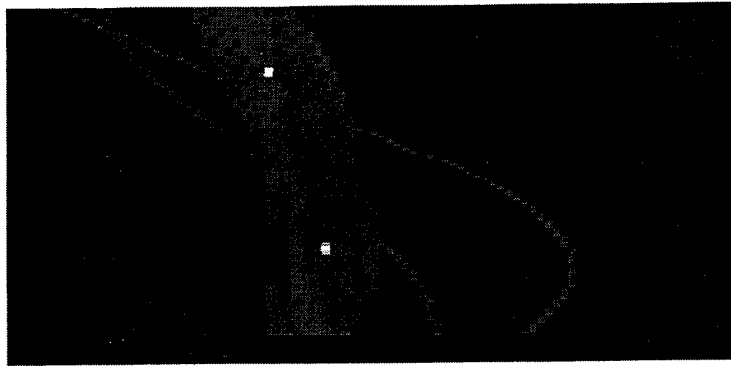


Figure 12: 2 line plus 2 point Hough Transform Output

#### (b) Timing Results

The timer was started just before the transform outer loop and was stopped immediately following the end of the outer loop. The timer, therefore, measured the time to process all 22 input points for all 90 angle hypotheses. What was not included in the timing tests was the initialization and setup, which included the building of the XCOS and YSIN tables and the zeroing of the histogram output table. It was decided that in any fielded implementation of this system, these setup times would not be a part of the acquisition-process-send output loop, and therefore, should not be a part of the critical timing measurements.



For several runs of the 22-point, 90 angle test, the timer measured between 26600 and 26750 timer ticks [the variation is due to the intrusive, but small, overhead functions of the MegaImager demonstration systems PC interface overhead - the software under tests uses exactly the same number of cycles every time that it is run]. Using 26600 as the measurement least effected by the MegaImager demonstration software, the number of instruction cycles is estimated to be 53,200 cycles. Since the inner loop is run once for every input point-angle pair, this measurement must be divided by (22 points x 90 angles =) 1980 iterations to get the inner loop timing. The inner loop is, therefore, estimated to take:

$$53,200 / 1980 = 27 \text{ cycles per inner loop iteration}$$

### (c) Timing Analysis - DRAM Page Faults

Though this number is somewhat higher than anticipated in the original proposal, it is easily explained when some details of the MegaImager board are considered. To provide the most RAM with lower cost, the MegaImager incorporates inexpensive and slow "page-mode DRAM" memory devices. With page-mode DRAMs, the access time is relatively fast (100 nsec or 2 instruction cycles) when reading or writing to sequential locations, but is very slow (250 nsec or 5 instruction cycles) when reading or writing to widely separated locations (more than 4096 addresses apart). When the last location accessed was not within 4096 addresses, a 'page fault' occurs, with the resulting 5 instruction cycle (250 nsec) penalty.

Many image processing applications, which process images in raster-scan order, never bump pointers more than one location at a time [thereby staying within a page 4095/4096ths of the time]. With a table-based Hough Transform, however, the XCOS and YSIN tables are accessed in each iteration of the inner loop. The tables are also too large to both fit within a DRAM's 4096 byte page. These tables are located on the 'C40's Global Bus, which is used for nothing else in the MegaImager demonstration software.

The output 2-D histogram is located in a different bank of page-mode DRAM's on the 'C40's Local Bus. The DRAM on the Local Bus is used for many things, including all data and executable code storage. No data is accessed with the inner loop (when the compiler optimization option is selected), but the executable code must be accessed. The MegaImager environment, however, leaves the 'C40's instruction cache enabled - which means that only the first pass of the code will require program accesses the code on the Local Bus (all subsequent iterations will use the internal cache - the inner loop is only 12 words long and the cache is 64 words long). In this analysis, the Local Bus DRAM can be considered to only be accessed for histogram output writes within the inner loop.

The 2-D histogram is accessed in adjacent lines from one inner loop iteration to the next. The length of a line is determined by the integer precision required by the user for the *rho* calculation. For this test, the resolution has been set to 100 equally spaced values, so the line length is 100 words. Since the histogram contents are represented as 16-bit data, this translates into 200 bytes. So from one iteration to the next, the adjacent accesses will be between 398 and 2 bytes apart, with a mean of 200 bytes. The expected frequency of a histogram output page fault is, therefore,  $(200 / 4096 =) 1/20^{\text{th}}$  or only 5% of the time.

The cost of DRAM page faults is:

$$\begin{aligned} & (2 \text{ read page faults} \times 5 \text{ cycles}) + (1 \text{ write page fault} \times 4 \text{ cycles} \times 5\%) \\ & = \mathbf{10 \text{ cycle page fault per inner loop iteration (when there is NOT a page write fault)}} \end{aligned}$$

This cost is board-specific and should not affect any practical implementation of the ARDD system. Since this application demands speed and has modest memory requirements, SRAM should be used which would eliminate the timing penalties associated with page-mode DRAMs. The remaining analysis, therefore, ignores this 10 cycle penalty.

#### **(d) Timing Analysis - Processing**

A review of the results so far shows that there are: 27 cycles/inner loop - 10 cycles for page faults = **17 cycles/inner loop** of total processing

A zero-overhead "block repeat" capability of the 'C40 processor is used to speed up inner loops like this one, so the remaining time must be allocated to a combination of outer loop overhead and inner loop processing. The inner-most loop happens 45 times (half of the angles) for each outer loop iteration (the points loop). The outer loop costs (by inspection) approximately 35 cycles per iteration, or the equivalent of about  $(35/45)$  1 cycle per inner loop iteration. It is unlikely that any better timing performance could be achieved. So the cycles due exclusively to the inner loop processing are: 17 cycles/inner loop - 1 cycle for outer loop overhead = **16 cycles/inner loop only**. Within the inner loop there are 12 instructions. The 'C' code used for the inner loop is shown here:

```

    for (angle=0 ; angle<45 ; angle++) {          /* LOOP 45 times
*/
    rho = *p_ysin++-*p_xcos++;
    ++(p_histogram[(int)((rho+MAX_RHO)*SCALE_RHO)]);
    p_histogram += 100;
    }                                              /* ENDLLOOP */

```

The assembly language (which is effectively equivalent to machine code in the 'C40 processor) generated by the optimizing compiler for that 'C' source is shown here (with the 12 inner loop instructions shown in bold type):

```

    LDI  44,RC
    RPTB L39

    SUBF  *AR5++,*AR2++,R9          ; rho = *p_ysin++-*p_xcos++;
                                     ;
                                     ;
    MAX_RHO)*SCALE_RHO)]);
    ADDF  R2,R9,R0
    MPYF  R10,R0
    FIX   R0,R1
    NEGF  R0
    FIX   R0
    NEGI  R0
    LDILR0,R1
    ADDI  R1,AR4,AR0
    ADDI  1,*AR0,R0
    STI   R0,*AR0

L39: ADDI 100,AR4                  ; p_histogram += 100;

```

The 12 inner loop instructions are dominated by the code which scales the floating point *rho* calculation into an integer index into the histogram. The floating point *rho* calculation requires only 1 instruction (though, due to the board's slow DRAM, it takes most of the cycles), but the rescaling and float to integer conversion requires 8 of the 12 instructions (approximately 8 cycles). Though a detailed analysis has not been attempted, it appears that the XCOS and YSIN tables could be pre-scaled to reduce this rescaling cost to nothing. Though carefully avoided here, there is also no reason that negative indices should not be allowed, as long as carefully "p\_histogram" pointer maintenance is performed and the histogram data space is allocated correctly. This could make the inner loop cost: 16 cycles - 8 cycles for rescaling *rho* = 8 cycles/inner loop as projected

### (e) Summary of Timing Analysis

The measured timing of the Hough Transform was 27 cycles per inner loop. Of those 27 cycles, the analysis has produced the following breakdown:

**Table 3 - Summary of Timing Results**

1 cycles	of outer loop overhead*
8 cycles	of critical processing*
8 cycles	of rescaling rho (can possibly eliminate)**
10 cycles	of DRAM page faults**
27 cycles	of measured inner loop timing

\* - These 9 cycles are projected to be unavoidable.

\*\* - These 18 cycles can be eliminated in a fieldable system.

#### (f) Conclusions

The Hough Transform execution time per image can be calculated using the following formula:

$$A * P * I * C = \text{Execution Time per Image}$$

where, A = Number of Angles over 180 degrees (typically 90 or 180)  
P = The Number of Points to Process (2500 for 1% of 512x512)  
I = The Number of Inner Loop Cycles (from 9 to 27, see Table 1)  
C = 'C40 Instruction Cycle Period (50 nsec for 40 MHz, 40 nsec for 50 MHz)

There are several permutations of the variables in the timing equation which could be of interest. The number of angles will probably increase with the number of points, recurring better angular resolution to pick the lines out of the noise. The number of points may increase if the image size increases or the sensitivity requirement increases. There is also a 50 MHz 'C40 available, which could affect the cycles per second. To explore these permutations the following table presents several possible scenarios of possible interest:

**Table 4 - Summary of Projections and Results**

A, Num of Ang.s	P, Num of Points	I, Num of Cycles	C, Cycle Period	Time per Frame	Max Frame Rate	umber of 'C40's for 30 frames/sec
90	2500	27	50 nsec.	<b>**0.303 seconds</b>	3.2 frames/sec.	10 (40 MHz)
90	2500	17	50 nsec.	<b>0.191 seconds</b>	5.2 frames/sec.	6 (40 MHz)
90	2500	9	50 nsec.	<b>0.101 seconds</b>	9.8 frames/sec.	3 (40 MHz)
90	2500	9	40 nsec.*	<b>**0.081 seconds</b>	12.3 frames/sec.	3 (50 MHz)
180	2500	9	40 nsec.	<b>0.162 seconds</b>	6.2 frames/sec.	5 (50 MHz)
180	5000	9	40 nsec.	<b>0.324 seconds</b>	3.1 frames/sec.	10 (50 MHz)
180	25000	9	40 nsec.	<b>1.62 seconds</b>	0.6 frames/sec.	50 (50 MHz)

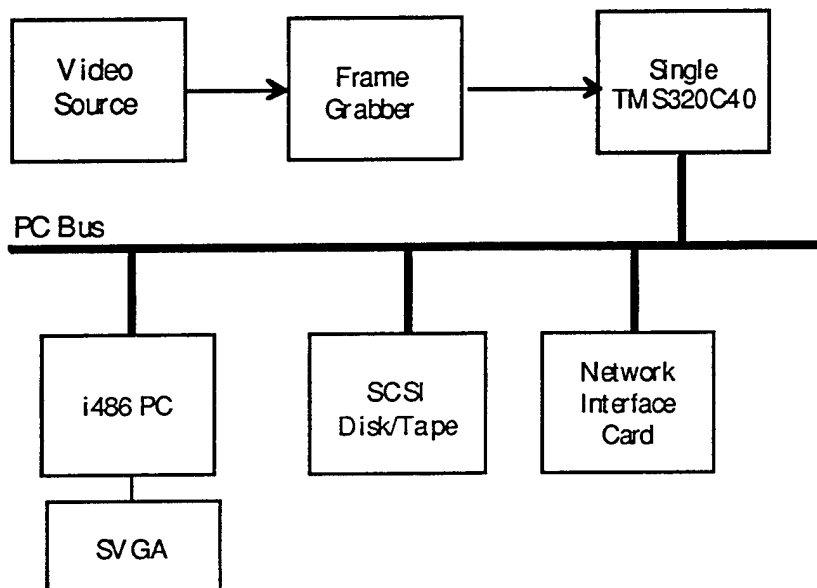
\*\* 0.3 sec. was extrapolated from the MegaImager timing measurements (with 2500 rather than 22 pts)

\*\*\* 81msec. should be achievable with three 50 MHz 'C40s, 1% thresh., and fast inner loop and HW

## 5.4 Closed Loop Realization of the ARDD

### 5.4.1 System Setup

The Hough Transform was integrated with other components of the system to provide a closed loop realization (Figure 13) of the ARDD using a single TMS320C40 board to perform detection of orbital objects recorded in video tapes.



**Figure 13 - Closed Loop Realization of the ARDD**

The closed loop algorithm was partitioned into several components and processed in sequence: frame grabbing, mean/variance computing, data preparing, Hough transforming, and result displaying. Video images were continuously digitized to 480x640-pixel frames and stored in the image memory of the DSP board. Pixel intensity means and variances were computed using six (6) moving frames and were used to establish normalized threshold. The data preparation produced a data array of pixels in which each has intensity (brightness) higher than a normalized threshold. The computation flow is simplified as follows:

- 1) Initialize the system
- 2) Acquire several frames and estimate noise
- 3) Set accumulator  $A(r,a) = 0$
- 4) Acquire a frame
- 5) Select all pixels (limit to 3000) that exceed noise + threshold
- 6) Determine  $r, a$
- 7) Increment accumulator  $A(r,a) = A(r,a) + 1$
- 8) Repeat steps (5) to (6) for all selected pixels
- 9) Peak in  $A(r,a)$  gives the line gradient and intercept
- 10) Update noise
- 11) Scale and plot  $A(r,a)$
- 12) Repeat steps (3) to (11) for all incoming frames

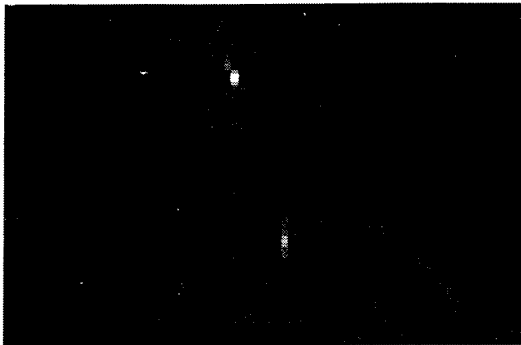
By setting the number of pixels exceeding a desired threshold at 3000 pixels (or 1 percent) and the number of angular increments at 90, the result has shown that a single 40-MHz TMS320C40 chip could process three (3) images of 480x640 pixels per second. This performance was lower than our original estimation of six (6) images per second because: (a) the DMA capability of the chip was not utilized in transferring data from the frame grabber to on-board memory, (b) the on-board memory was slower (2-5 wait states) than our original assumption (1 wait state), (c) the requirement of scaling results and interacting with the PC on an ISA bus was added.

#### 5.4.2 Synthesized Data

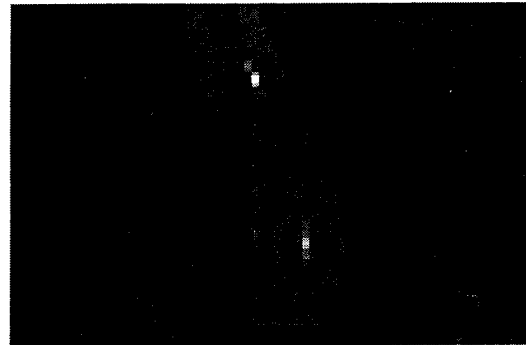
Several sets of synthesized data were generated and applied to the closed-loop environment to verify the operation and accuracy of the setup. Data were generated by the DSP and deposited into the frame grabber buffers.

As shown in Figure 14(a), when two lines and two discrete pixels were applied, the output of the Hough transform had two (2) peaks at proper locations as expected. When two discrete pixels were removed, the peaks did not alter while two dim lines disappeared as in Figure 14(b).

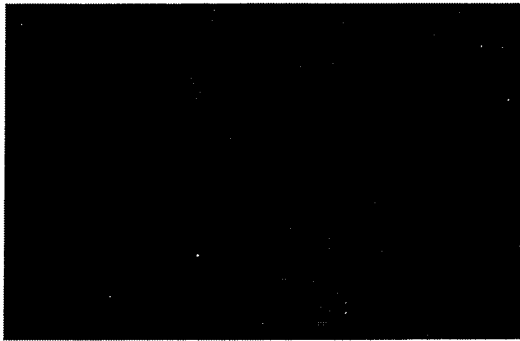
To continue the verification, four discrete points were applied to the system. In this case, the Hough transform did not result any peak, as shown in Figure 14(c). This conveyed the fact that there was no dominating line formed by four points. On the other hand, as shown in Figure 14(d), when two parallel lines with a different number of points were applied, several peaks were shown in the parameter space. It was confirmed that the highest peak, indeed, associated with the fifteen-point line and the next peak corresponded to the ten-point line.



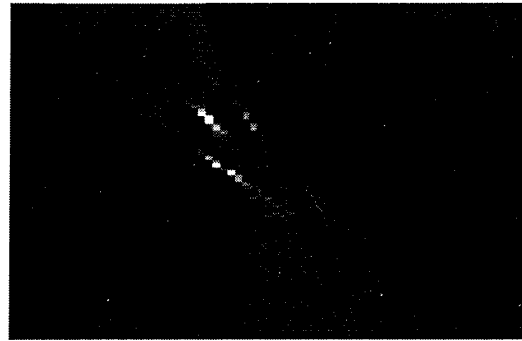
(a) 2 lines at  $-45$  and  $+45$  degrees. 10 points per line. 2 additional points at  $(600,400)$  and  $(600,0)$



(b) 2 lines at  $-45$  and  $+45$  degrees. 10 points per line.



(c) 4 points at  $(100,100)$ ,  $(100,200)$ ,  $(200,100)$  and  $(200,200)$



(d) 2 parallel lines. One line with 10 points and another line with 15 points.

**Figure 14: Synthesized Data**



### 5.4.3 Pre-recorded Data

Video tapes provided by PL/GPIM were played back at a speed of three frames per second. This adjustment was needed so that the closed-loop operation appeared as if it was in real-time. Each frame is digitized and cropped to fit a 480x640 array of pixels. For the sensor in question, the width of the digitized frame corresponds to the field of view of approximately one degree. At orbital speeds, an object at 500 km altitude travels the full width of the frame in about one second. On the other hand, stars which travel at speeds less than 0.004 deg/sec are expected to stay stationary from frame to frame.

An instance of raw image captured by the frame grabber is shown in Figure 15. Its column and row profiles are shown in Figure 16. It is noticed that stars are shown as bright objects and background noises from equipment are scattering in the captured image.



Figure 15: An Instance of Raw Image

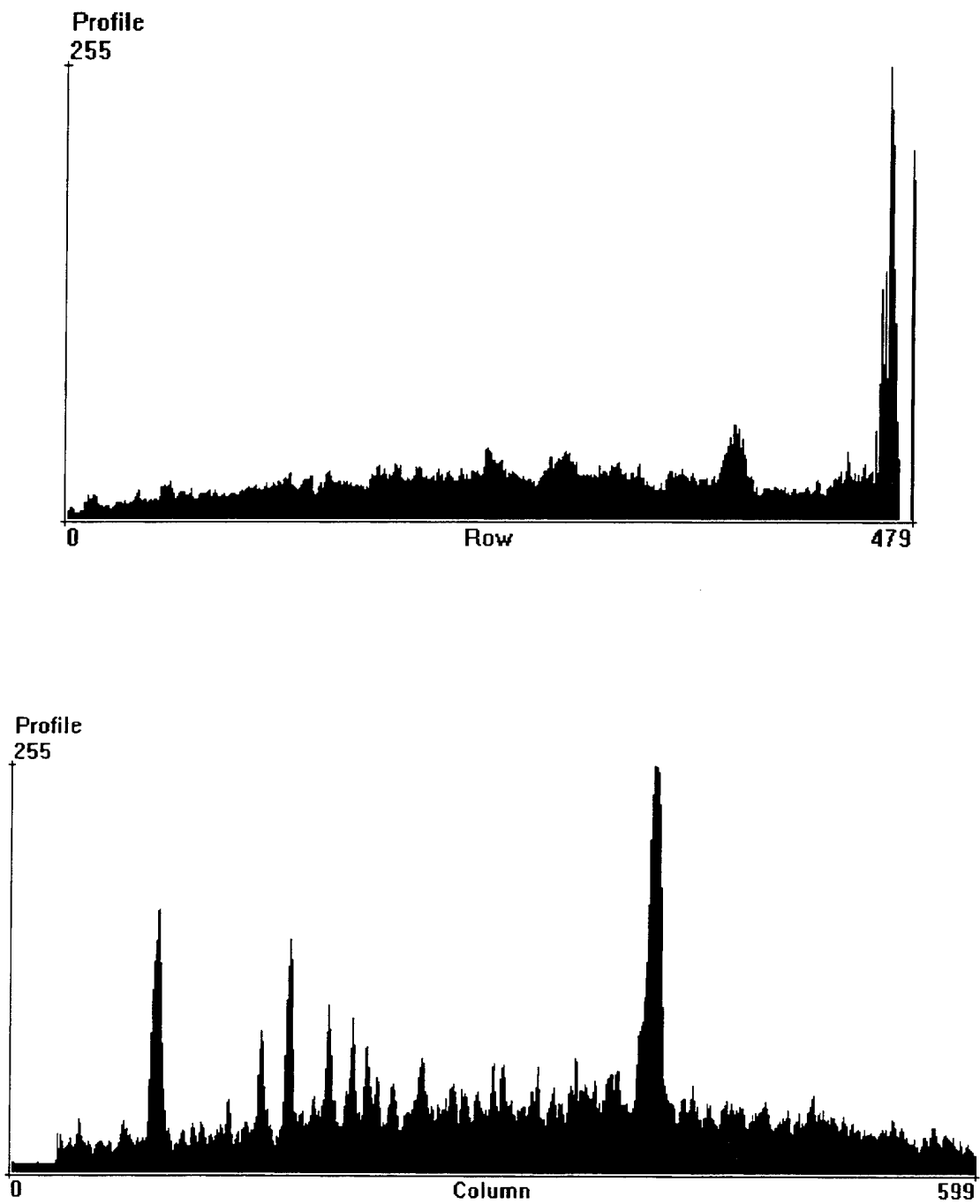
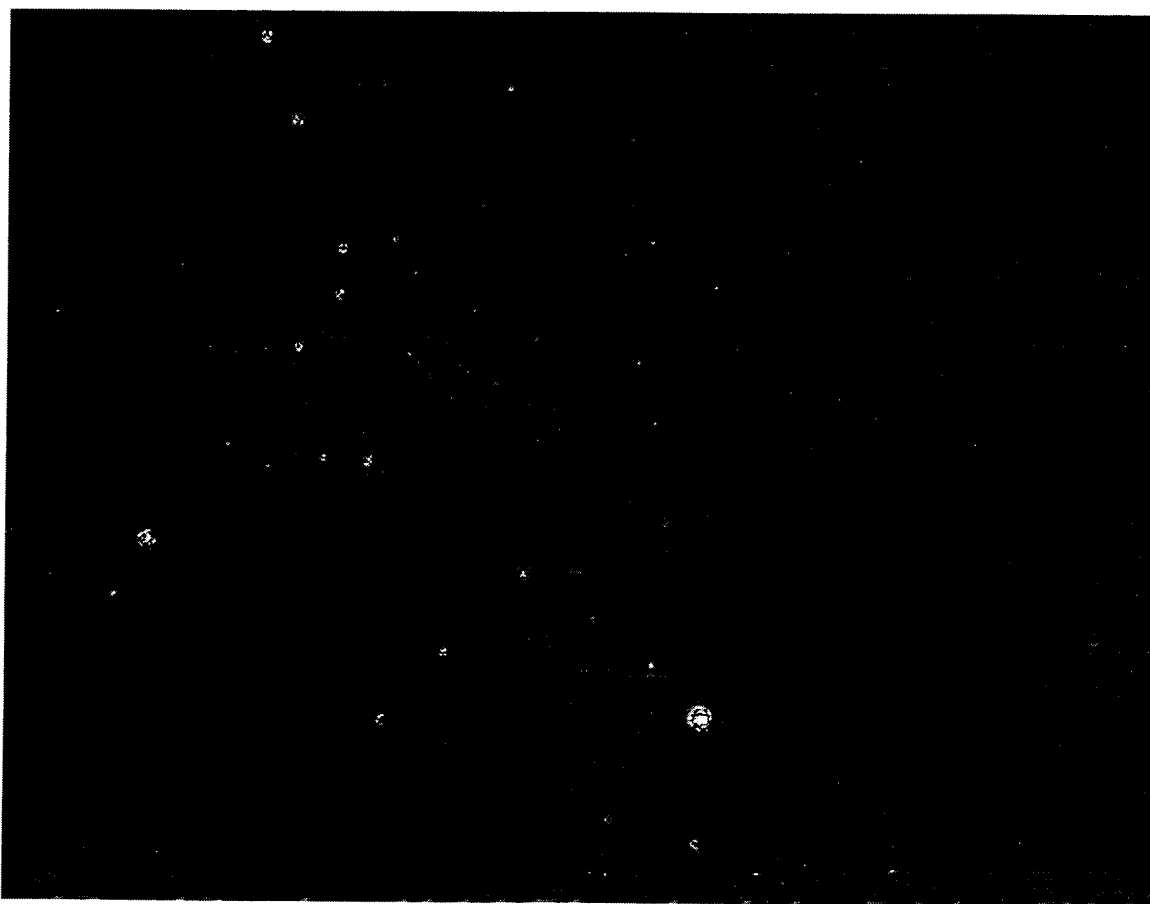
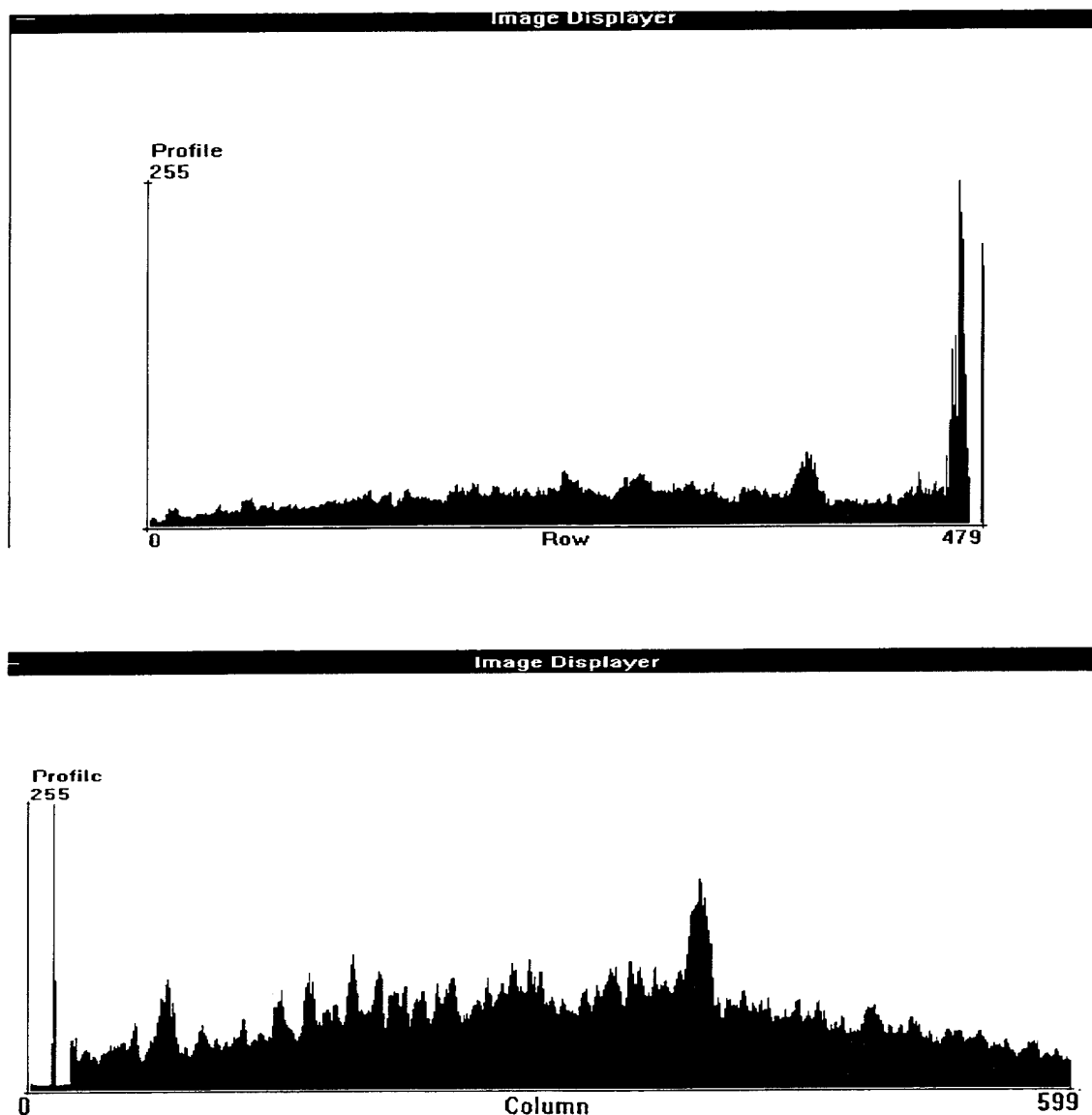


Figure 16: Row and Column Profiles of Raw Image

Noise was estimated for every pixel by using six-frame moving average technique. Objects whose intensity does not exceed the estimated noise by more than the pre-computed standard deviation are rejected. The effect of this step was to ensure that stationary objects were filtered out. However, this filter did not work perfectly because stars flicker (breathe) and wander due to atmospheric turbulence. Residual pixels were left behind near the original position of the star and are evident in the filtered frame (Figure 17). These residual pixels contributed substantially to the performance of the system and will be discussed later.



**Figure 17 - Estimated Noise**



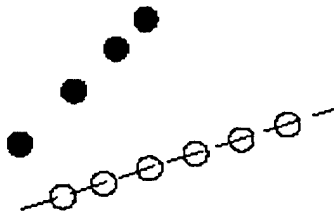
**Figure18: Row and Column Profiles of Estimated Noise**

The digitized pixel values were quantized such that the pixels with relative brightness values in the 99-percentile range were retained for further processing. In other words, the thresholding/quantization retained 1% of the pixels. This step eliminated most of the frame including the dark spaces between the stars and debris (moving) objects. For the Hough Transform detection system to work, debris (moving) objects have to be bright enough (in relative to estimated noise) to be included in this 99-percentile range. This threshold was arbitrarily chosen to benchmark the system throughput and will be adjusted to an optimal value in the operational system.

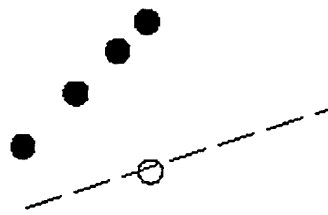
#### 5.4.4 Analysis and Conclusion

There is no convenient way to present the results in a graphical form, yet the analysis reveals important information impacting the final implementation. The thresholded/quantized frames were analyzed by Hough Transform with the following parameters. Hough Transform processing was implemented with 90 angle values or a 2 degree angular resolution. There were 100 distance bins that allowed us to determine the position of the detected object to 1% of the screen width or 0.01 degree of the sensor's field of view. Because of the 2 degree angular resolution, stationary residual "noise" affected the result as follows. Inspection of data showed that in a given thresholded/quantized frame there were always groups of stationary points that happened to stay on a line (to within 2 degree). These groups of "linearized" points outnumbered those contributed by the moving target. As a result, thresholding the Hough Transform frame will not locate the moving object.

For example, Figure 19(a), a moving object forms a straight line with six points over a period of time. In this case, the highest accumulator of the Hough Transform reveals the gradient and intercept of the line formed by the moving object. However, when a frame is acquired at some instance of time, the moving object is seen as a single point. In this case, Figure 19(b), the highest accumulator of the Hough Transform corresponds to the line formed by four stationary objects. Furthermore, four additional lines with two points per line are formed by the stationary objects and an instance of the moving object. The gradients and intercepts of these lines are contained in the next lower accumulators. Therefore, if time integration of frames is not used, the moving object is always masked by the line formed by stationary objects and will not be detected.



(a) Time integration of moving object (clear dots) and stationary objects (e.g. stars, solid dots).



(b) At an instance of time, stationary objects still form a line while moving object is seen as a singular point.

**Figure 19 - Moving Object vs Stationary Objects**

During the closed-loop testing and analysis, many played back tapes contained groups of stars aligning as described above. Groups of four linearized star points were not uncommon and were mistakenly identified as moving objects.

Although the system was not built to have the full capability of an operational detection system, the difficulty has been briefed to PL/GPIM. Additional working sessions were held (May & June 1995) to identify appropriate short and long term directions. Three tentative approaches were identified.

The first approach is to perform time integration of image frames prior to Hough Transform. Based on the assumed speed of debris objects, as much as a full second of frames should be integrated to improve sensitivity. The integration time is equivalent to thirty frames in a real-time video system. This step warrants that the moving object contributes more points to an integrated frame than an average group of "linearized" noise points, henceforth referred to as the linearized sidereal noise. Finally, Hough Transform is applied on the integrated image frame and its output thresholded to identify the moving object.

The second approach is to apply time integration directly on the output frame of Hough Transform. Given the assumption that lines formed by stationary or sidereal objects are time-invariant or vary very slowly in time, their pixel accumulators and coordinates in the Hough frame do not substantially alter from frame to frame. Therefore, they can be determined and discarded during integration of Hough frames. In this approach, the Hough accumulators corresponding to the "linearized" sidereal noise have predictable coordinates and are obviously related to the star field at the given time.

A third approach can be taken to mitigate the contribution of stationary objects such as stars or imperfections in the imaging detector. In this approach, holes are drilled in the image frames before Hough Transform. A hole can be drilled about the point identified to be stationary. Because the star field changes slowly and predictably, the nominal coordinates of these points are known in real time. If each hole is sufficiently larger than the area over which a typical star is expected to "wander" and "breathe" due to atmospheric turbulence, the stationary noise points can be removed effectively. The difference between this approach and the second one is that a large number of drilled holes are required as compared to a smaller number of masked-out points required by the second approach.

## **5.5 Future Development Efforts**

### **5.5.1 Approaches**

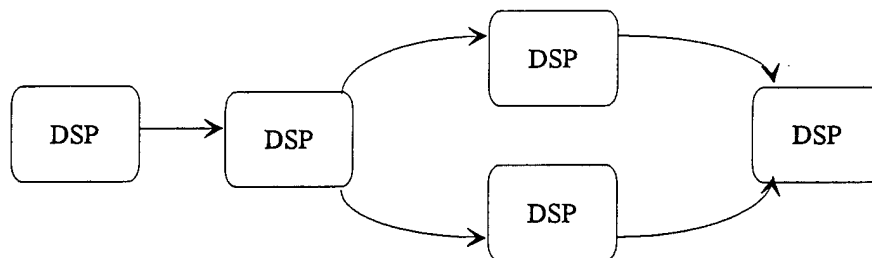
The future development efforts are based on two distinctive approaches. The first is to build the system based on the current Digital Signal Processor (DSP) technology (TMS320C40) while the second path is to employ advanced Multimedia Video Processor (MVP) technology (TMS320C80).

To implement the system for real-time operation (30 frames per second), an initial configuration consisting of five (5) Texas Instruments TMS320C40 DSPs with faster on-board memory will be used. This configuration provides more than 200 MFlops of processing power and will fulfill the current computing requirement for 512x512-pixel image with the number of pixels exceeding a desired threshold is 2500 pixels (or 1 percent) and the number of angular increments is 90.

Alternatively, a single TMS320C80 MVP chip is used instead of multiple TMS320C40 chips. With a throughput of over two billion RISC-like operations per second and an architecture tailored to imaging applications, the MVP is a better choice for future expansion. Initial benchmark from Texas Instruments has shown that the throughput of an MVP is about ten (10) to twenty (20) times of that of a DSP.

### **5.5.2 TMS320C40**

Using the throughput estimation in Section 4, parallelism can be achieved to meet the real-time requirement of 33.3 msec per frame by connecting five (5) TMS320C40 DSPs. One DSP is allocated to compute the variances during non real-time mode and computes means during real-time operation. The variances computed in non real-time mode are then adjusted to a desired probability of detection to establish 512x512 pixel thresholds. Another DSP is allocated to normalize and compare the incoming pixels to the pre-determined pixel thresholds to generate data array for the Hough Transform. Three additional DSPs are allocated to perform Hough Transform. The first is for the transformation of the first 90 angular increments; the second is for the transformation of the next 90 angular increments; and the last is for combination the results produced by other two DSPs. Figure 20 shows the DSP connection topology in which each DSP connects to others using communications ports.

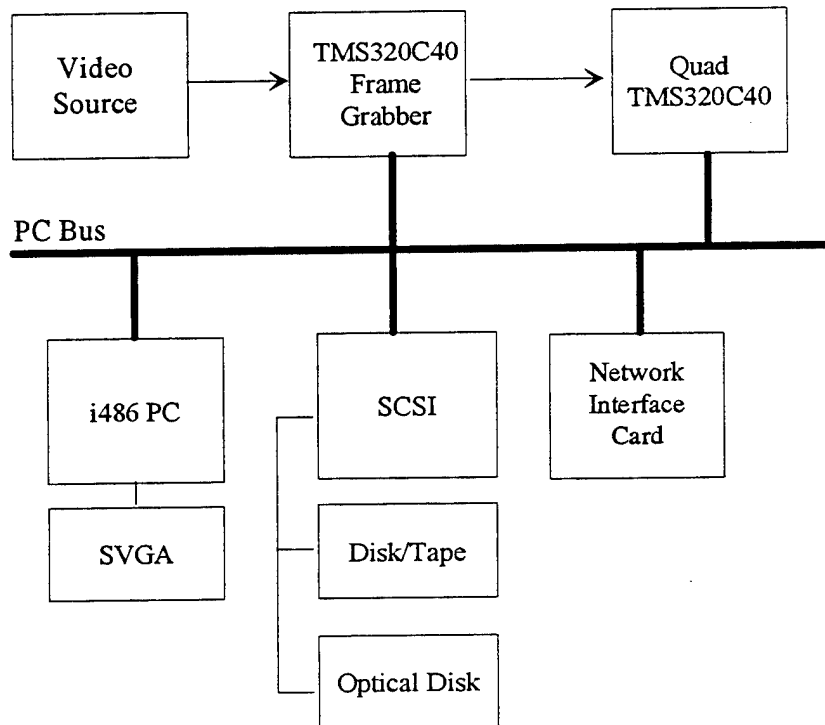


**Figure 20: Topology of DSPs**

The frame grabber digitizes an incoming image into two alternate memory buffers, referred as buffer 0 and buffer 1. In this scheme, data is being deposited in one buffer while the DSPs process data in another buffer. Suppose that 512x512 pixels of an image are available in the buffer 0 at the end of dwell N. The first part of the data preparation (DP0) will complete the mean computation at the end of dwell N+1 and the second part of the data preparation (DP1) will complete threshold comparison at the end of dwell N+2. The Hough Transform (DP2 and DP3) of the first and second 90 angular increments will take place at the beginning at dwell N+3 and conclude at the end of this dwell. Combination (DSP4) of both 90 angular increments starts at the beginning of dwell N+4 and completes at the end of this dwell. The final information about the image at dwell N is available at dwell N+4. This equals a total delay of 133.3 msec.

A high bandwidth bus is not required because transferring of imaging data is carried by the DSP communication scheme. The system bus merely provides power distribution and carries system control command. Two commonly used buses -- PC and VME -- are considered and both can accommodate the technical requirements of the system. PC-based hardware, as shown in Figure 21, is recommended mainly because of lower cost of PC-based components.





**Figure 21: PC-based System Hardware**

The Frame Grabber board incorporates high speed image acquisition with simultaneous processing using Texas Instruments TMS320C40 DSP running at 40 Mhz or higher. It is available with a single analog video input daughter board, capable of a digitization rate of up to 20 MSamples per second. The Frame Grabber board also has more than 16 Mbytes (zero or 1 wait state) of on-board image memory and two (2) TMS320C40 communication channels reserved for inter-board communications.

The Quad 'C40 board consists of four (4) TMS320C40 DSP running at 40 Mhz or higher. The board has more than 16 Mbytes of zero wait state static RAM on-board or 4 Mbytes per 'C40. Connection of on-board 'C40 DSPs is made possible via TMS320C40 communication channels. The Quad 'C40 architecture allows unlimited number of boards to be connected for expansion.

The host has two primary roles: it provides a complete environment for development effort and acts as system controller during operation. For development, the host consists of several software tools to support the development of host software and target software. As system controller, the host consists of utility library to download executable images to and upload data from DSPs in order to control the DSPs and monitor the system status. In addition, the host also displays the final image on its graphical display or records detection information to the system hard disk.

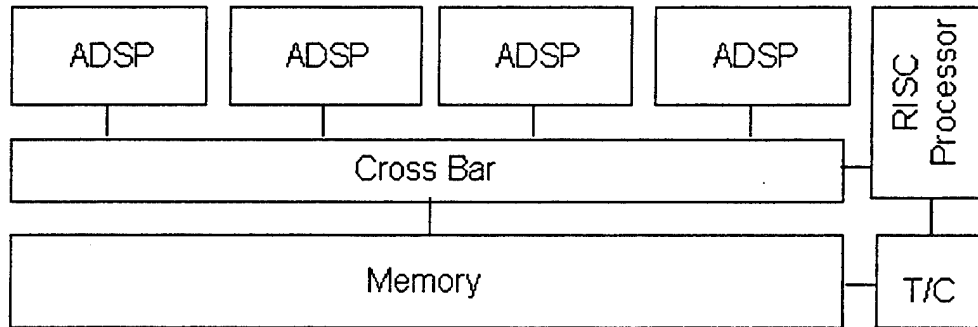
The system storage consists of a SCSI board interfacing to the system disk, optical disk, and tape backup units. The initial storage size is about one (1) Gbytes for software development and digital data recording. Storage capacity can be increased by attaching additional units.

The network interface card consists of an Ethernet link and a modem link. The Ethernet link will be used for local area communication during development. The modem link will be used to communicate with other remote subsystems. The modem link also be used to link to other ARDDs located at different geographical sites.

### **5.5.3 TMS320C80**

The TMS320C80 integrates onto a single integrated-circuit five powerful, fully programmable processors, a sophisticated DMA (direct memory access) controller with a DRAM, SRAM, and VRAM external memory interface, 50K bytes of SRAM, and video timing control.

As shown in Figure 22, a simplified block diagram of the 'C80, four of the five processors are identical. These are advanced digital signal processors (ADSPs) that have hardware to support multiply-intensive processing, bit-field-intensive pixel manipulations, and bit-field intensive operations. Each of the ADSPs is capable of performing many RISC-equivalent operations in a single cycle. The fifth processor is a 32-bit RISC (reduced instruction set computer) master processor with a high performance IEEE-754 floating-point unit.

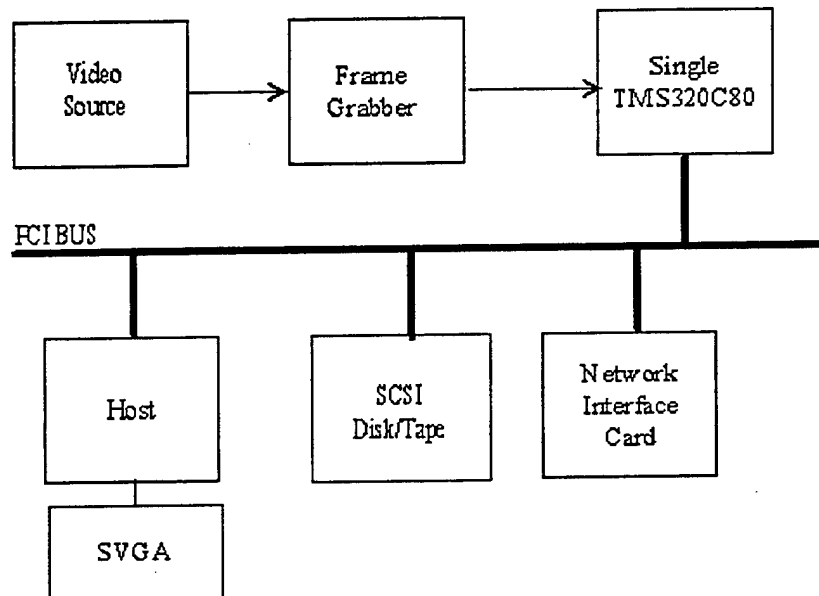


**Figure 22: Simplified Block Diagram of TMS320C80**

In addition to the fully programmable processors, the transfer control (TC) is an intelligent DMA controller that manages all memory traffic. The TC performs packet transfers that move data between on- and off-chip memory. These packet transfers include instruction-cache and data-cache servicing, as well as complex programmable byte-aligned array transfers that can include X/Y or linear addressing of the source or destination array.

The 'C80 is capable of performing the equivalent of over two billion RISC-like operations per second. In some applications, a single 'C80 can do the job of over ten of the most powerful DSPs or general-purpose processors previously available. During each second of processing, the 'C80 can move 2.4 Gbytes of data and 1.8 Gbytes of instructions within the chip, plus 400 Mbytes of data to off-chip memory.

For the ARDD project, a single 'C80 board can be used to replace the quad 'C40 board while preserve the system architecture as shown in Figure 23. However, with the availability of PCI bus architecture and Microsoft Windows NT operating systems, the host computer does not need to be an Intel x86 processor.



**Figure 23: PCI-based System Architecture**

There are several ways to partition the ARDD processing into tasks and to assign tasks to the master processor and four ADSPs. These methods are all feasible thanks to the cross bar and shared memory architecture of the 'C80.

The first method, function allocation, is very similar to the TMS320C40 scheme. Functions performed by each 'C40 chip will be mapped into an appropriate ADSP of the 'C80.

The second method, frame allocation, is to assign the work based on frame sequence. For example, frame  $n$  is to be processed by ADSP 0, frame  $(n+1)$  is to be processed by ADSP 1, frame  $(n+2)$  is to be processed by ADSP 2, and frame  $(n+1)$  is to be processed by ADSP 3. This will leave the master processor to perform final decision making.

The third technique, pixel allocation, is to partition and assign the work in the level of pixel or group of pixels. For example, when a frame arrives, the master processor may partition the image into some number of sub-images and then command the ADSPs to perform the job of processing the sub-images.

## 6. Spatial Density Modeling:

### 6.1. Objectives:

We performed a simulation of expected values of potential sightings of space debris objects based on a known compilation of starting values and then extrapolating the orbital parameters to the next set of values over a selectable time interval (normally two days). We wanted to see how the expected values we compiled correlated with the actual values received from the participating sites, AMOS (Hawaii) and Socorro, NM. The procedural flow is shown in Figure 24.

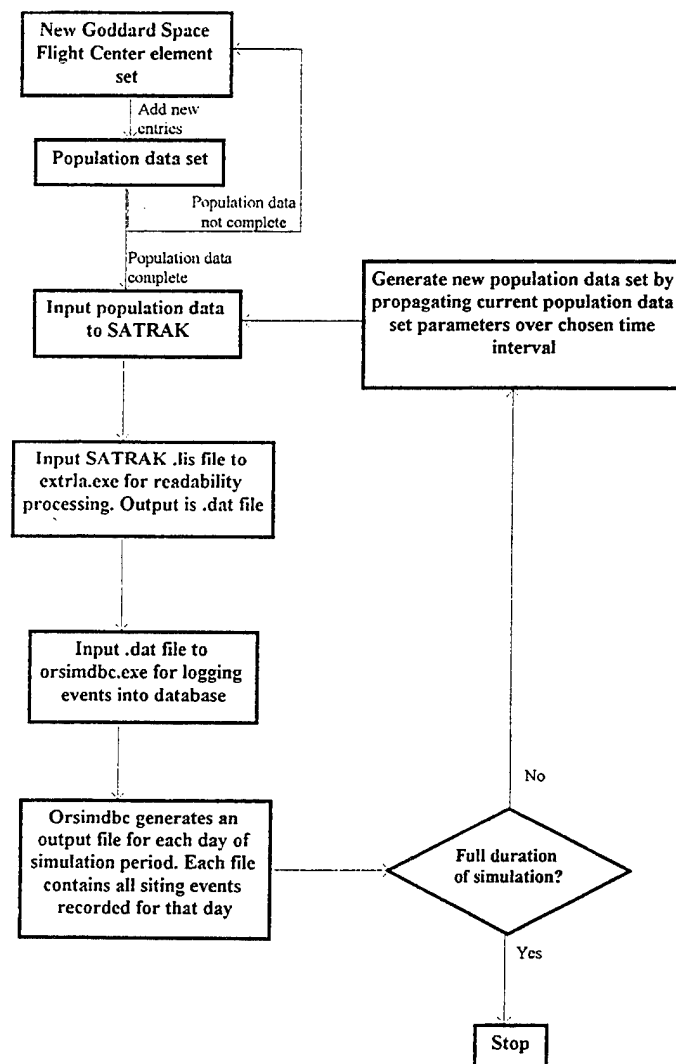


Figure 24: Procedural flow of spatial density modeling

## 6.2 Two-line element sets:

The first task was to compile a good sampling of the tracking data output by Space Command. We wanted to achieve a representative sample of the population that we were dealing with. Goddard Space Flight Center posts "two line element sets" for acquisition by research sites that have a need for such information. These data sets are posted on a VAX computer and accessed by File Transfer Protocol (FTP) for downloading.

In order to assure a continuing and non-interrupted sequence of data sets, we automated the acquisition process by a series of UNIX c-shell scripts. The data sets were posted on Monday, Wednesday, and Friday of each week. At 6 PM on those days, the UNIX daemon process would wake up and log on to the VAX computer at Goddard, retrieve the current data set and log off. The automatic process would then compress the downloaded data set and go back to sleep.

## 6.3 Software Components Setup:

The population data was compiled by starting with an arbitrary data set and moving forward in time adding those debris objects that were not already in the set. A series of applications have been written for the orbital debris statistics simulation. The simulation is coded in the object-oriented language, C++, which gives us greater flexibility, reusability, and robustness. A satellite tracking program, SATRAK, in conjunction with the element sets propagates the orbits of each debris object to some later time for observation, further tracking, or analysis. In the orbit simulation, we start with an initial element set's values and propagate the parameters that would change under normal conditions (dynamic parameters) for each debris object. After the first iteration, we have another element set of all the debris elements contained in the first set. The new set's elements have modified values for each of their dynamic parameters representing the changed state of each debris object. This process consisted of adding two days to the epoch time and modifying mean anomaly and mean motion to reflect the new orbital point in time. Using the March 5, 1993 element set as the initial population, each subsequent set was derived from the initial set by updating (modifying) certain parameters contained in the set. At the end of the process, we have two-line element sets for all the debris objects contained in the initial set, propagated over any interval we choose. The next step of the simulation is to input each of the element sets into the satellite tracking program. The tracking program outputs one list file for each detector site of interest. The list file contains the debris objects and the parameters necessary to observe each one as they pass within viewing range for the user-designated site. The list file's parameters give the necessary information, in raw form, to compile distribution statistics on the debris population. The list file, in turn, is processed by extrla.c (extract look angles). The output of

extrla.exe (.dat file) is a more readable file than the aforementioned list file and presents the parameters (altitude, azimuth, and time) necessary for locating the debris objects during a morning or evening viewing session. Extrla's output file is processed further by the program orsimdbc.exe (orbit simulation database create). Orsimdbc.exe extracts information from the two-line element file and extrla's .dat file, for each object that can be observed from a particular site, and appends that potential observation to a file named for the day that the observation occurred. The program appends current data to previous data in each day file if any exists, otherwise it creates the file. Preliminary results were analyzed for a twenty-day period to establish the validity of the model. The analyzed results agree with expected values.

#### 6.4 Data analysis:

We received space debris tracking data from the AMOS site in Hawaii and the Socorro site in New Mexico. The purpose of this data was its use in comparisons between actual tracking and the spatial density simulation mentioned previously. We wrote several programs to massage this data. The first program processes data received from space debris observation and tracking sights, AMOS and New Mexico. The textual data are in individual files, each representing a night's observations. The file name encodes the type of file (coverage, B3, brightness, or element set), the year, and the location of the observations. The output file names combine these fields to identify the file type and location and are then used as inputs to a data analysis program. The second program performs essentially the tasks of the first except on a different format. That program also takes a number of text files of arithmetic data adding, modifying or eliminating fields in each file and then combines each into a file that is used as input to the same analysis phase.

The simulation was terminated after the 42nd update to the original element set. Data analysis was carried out for inclination, semi-major axis, and eccentricity. Histograms were calculated for ten-day intervals from day 10 through day 80 for each of the parameters. Each of these graphs contained two plots for comparison: the simulation data and the population data. Additionally, for 20 day intervals starting from day 20 through day 80 a correlation coefficient was computed and plotted for each interval. The correlation coefficient gave a quantitative measure of the agreement between the simulation data and population data histograms at each 20-day interval. Generally, these plots were monotonically increasing and thus showed the convergence between the simulation and the population data.

However, there were some unexplained peaks in the computed data that we could not readily explain, and which seemed non-physical in nature. Unfortunately, the Keo software engineer who was working on this problem left Keo employment, so the project was placed on hold while the future direction of this simulation effort was evaluated with Air Force personnel.

## 7. Space Debris Population Simulation

### 7.1 Overview:

Keo Consultants was asked by PL/GPIM to develop a data base of valid Satellite Elements extracted from the NASA two-line element sets and to try and simulate a distribution of debris to look at expected observation rates for optical measurements at various observation sites. To address this question, NASA Element-Sets from Goddard SFC were downloaded on a regular basis and used these to create an appropriate database. This database was then used to predict look-angle observations for the location at Rattlesnake Mountain in Washington State for various fields-of-view.

### 7.2 Methodology:

The first step in this simulation was to look closely at the form of data presented in the NASA two-line element sets. It was found that the number of elements in each set varied widely, and there were many duplicate entries in sets for the same entry. Software was written to compile a list of all unique element numbers in a sequence of element sets.

The output listed below shows the results of this master compilation of unique element entries plus the first few 'output' entries for the MasterSet file. This output shows the input collection of NASA two-line element sets identified by the name **twoline-MM/DD/YY**, and also shows how many elements were contained in each element set. One can see the great variation from element-set to element-set below (varying from 2155 entries to 13722 entries):

### Master Set Build for Space Debris Propagator Model

**Version 1.0**                      **KEO Consultants**

Creates a master list of all elements found in the list of Element sets  
I.e. no objects are culled out

File Created: Mon Jan 23 18:55:33 1995

Input List Files:

```
***** Start of File List *****
Opened File #1: ::Element Sets:twoln-10/7/94
Opened File #2: ::Element Sets:twoln-10/10/94
Opened File #3: ::Element Sets:twoln-10/12/94
Opened File #4: ::Element Sets:twoln-10/14/94
Opened File #5: ::Element Sets:twoln-10/17/94
Opened File #6: ::Element Sets:twoln-10/19/94
Opened File #7: ::Element Sets:twoln-10/21/94
Opened File #8: ::Element Sets:twoln-10/24/94
Opened File #9: ::Element Sets:twoln-10/26/94
Opened File #10: ::Element Sets:twoln-10/28/94
Opened File #11: ::Element Sets:twoln-11/1/94
Opened File #12: ::Element Sets:twoln-11/2/94
Opened File #13: ::Element Sets:twoln-11/4/94
```



Opened File #14: ::Element Sets:twoln-11/7/94  
Opened File #15: ::Element Sets:twoln-11/9/94  
Opened File #16: ::Element Sets:twoln-11/14/94  
Opened File #17: ::Element Sets:twoln-11/16/94  
Opened File #18: ::Element Sets:twoln-11/18/94  
Opened File #19: ::Element Sets:twoln-11/21/94  
Opened File #20: ::Element Sets:twoln-11/23/94  
Opened File #21: ::Element Sets:twoln-11/28/94  
Opened File #22: ::Element Sets:twoln-11/30/94  
Opened File #23: ::Element Sets:twoln-12/2/94  
Opened File #24: ::Element Sets:twoln-12/5/94  
Opened File #25: ::Element Sets:twoln-12/9/94  
Opened File #26: ::Element Sets:twoln-12/12/94  
Opened File #27: ::Element Sets:twoln-12/14/94  
Opened File #28: ::Element Sets:twoln-12/16/94  
Opened File #29: ::Element Sets:twoln-12/19/94  
\*\*\*\*\* End of File List \*\*\*\*\*

29 files were opened in the Input List

Read File twoln-10/7/94 with 8119 two-line entries  
Read File twoln-10/10/94 with 4675 two-line entries  
Read File twoln-10/12/94 with 6590 two-line entries  
Read File twoln-10/14/94 with 2155 two-line entries  
Read File twoln-10/17/94 with 8343 two-line entries  
Read File twoln-10/19/94 with 3400 two-line entries  
Read File twoln-10/21/94 with 6327 two-line entries  
Read File twoln-10/24/94 with 6363 two-line entries  
Read File twoln-10/26/94 with 5483 two-line entries  
Read File twoln-10/28/94 with 4347 two-line entries  
Read File twoln-11/1/94 with 2558 two-line entries  
Read File twoln-11/2/94 with 5586 two-line entries  
Read File twoln-11/4/94 with 4386 two-line entries  
Read File twoln-11/7/94 with 8032 two-line entries  
Read File twoln-11/9/94 with 4388 two-line entries  
Read File twoln-11/14/94 with 4931 two-line entries  
Read File twoln-11/16/94 with 6734 two-line entries  
Read File twoln-11/18/94 with 3755 two-line entries  
Read File twoln-11/21/94 with 7839 two-line entries  
Read File twoln-11/23/94 with 4164 two-line entries  
Read File twoln-11/28/94 with 13772 two-line entries  
Read File twoln-11/30/94 with 3717 two-line entries  
Read File twoln-12/2/94 with 5064 two-line entries  
Read File twoln-12/5/94 with 6789 two-line entries  
Read File twoln-12/9/94 with 4159 two-line entries  
Read File twoln-12/12/94 with 7406 two-line entries  
Read File twoln-12/14/94 with 3856 two-line entries  
Read File twoln-12/16/94 with 5166 two-line entries  
Read File twoln-12/19/94 with 7911 two-line entries  
There were 11821 unique elements found in the above sets

Data Format: >> %05d %15s %15s %15s %15s<<  
 (One leading space, three spaces between parameters)

Start of Element Set Data Base:

SatNum	Oldest Set/Epoch	Newest Set/Epoch
00005	twoln-10/7/94 94276.15029512	twoln-12/19/94 94351.12881707
00011	twoln-10/7/94 94276.08468375	twoln-12/16/94 94349.83959258
00012	twoln-10/7/94 94276.07144168	twoln-12/19/94 94353.24209383
00016	twoln-10/7/94 94278.81814721	twoln-12/19/94 94352.13896339
00020	twoln-10/7/94 94275.96756845	twoln-12/19/94 94352.86104870

It was found that there were over 11,000 uniquely named objects included in these element sets for a three month period, whereas we would expect a number of tracked orbits on the order of 7,000. It was decided that many of these entries were false objects or duplicated objects that had been misidentified. An attempt to correlate this master database to the SATSIT reports was made, but again, these SATSIT reports seemed very inconsistent and not to produce a valid set of objects being observed.

It was decided to develop a criterion for a 'valid object' and use this criterion to create a new database of elements. The critereon used to produce this new database was that valid objects with consistent orbital elements should produce a consistent set of observations over time. Therefore, if we demanded that a valid object had at least two unique epoch entries in our collection of element-sets we should see a greatly reduced number of 'valid' objects.

Objects that were discarded would be objects that couldn't be tracked for more the selected time period and thus were probably false objects or falsey indetified objects. We expected to see the number of 'valid' objects converge on a number of around 7000 objects to agree with the NASA figures, and found that this indeed happened very quickly as we increased the minimum required time between two epoch entries.

The time period selected to create Keo's Database of valid orbital objects was **two weeks**, which produced a database containing 7051 objects. This was taken from the Master Database of collected element sets over a three month period. We make the assumption that if an object does not have two or more observations with a minimum spacing of two weeks over this three month period, it is probably not a valid orbiting object.

The following output is the header for this Database. Note that out of 11821 unique elements found in the Master Set, only 7051 were deemed by the above criterion to be valid objects. In this process, we also extracted an object's RCS value from the NASA RSC database, and puts it in the database for future correlations. The following header shows that of the 7051 elements stored in this database, only 6256 objects have corresponding RCS values.

## Database Build for Space Debris Propagator Model

Version 1.0 KEO Consultants

Creates a database that is a subset of the MasterSet  
based on the minimum entry period of 2 weeks

File Created: Tue Jan 24 15:06:36 1995

MasterSet File: ::#1 Master Set:MasterSet-1/23/95

\*\*\*\* DataBase successfully run: 11821 Unique Elements Found \*\*\*\*

\*\*\*\* There were 7051 valid orbits found \*\*\*\*

\*\*\*\* 4770 elements were culled out of database for insufficient orbital data

\*\*\*\* Insufficient orbital data defined as (Newest Epoch - Oldest Epoch) < 2 weeks

>>>> There were 11413 entries in the RCS File: nasa\_rcs.dat

6256 were found in this data set (88.7% of data set)

Entries without a corresponding RCS value are given an RCS = 99.9999

Data Format: >> %05d %7.4f %15s %15s %15s %15s<<  
(One leading space, three spaces between parameters)

Start of Element Set Data Base:

SatNum	RCS	Oldest Set/Epoch	Newest Set/Epoch
00005	0.3570	twoln-10/7/94 94276.15029512	twoln-12/19/94 94351.12881707
00011	0.7010	twoln-10/7/94 94276.08468375	twoln-12/16/94 94349.83959258
00012	0.7920	twoln-10/7/94 94276.07144168	twoln-12/19/94 94353.24209383
00016	0.6220	twoln-10/7/94 94278.81814721	twoln-12/19/94 94352.13896339

From this database, a listing of these objects was sent to PL/GPIM. The listing contained the objects orbital parameters and also a correlation with the NASA RCS parameters for that element which predicts the reflectivity of the object. This information was used for modelling purposes by PL/GPIM. The following output shows this file's header and the first few objects. The Apogee and Perigee of the orbits are determined from the Mean Motion and Eccentricity taken from the orbital elements. (The last valid NASA element set is used for each element's orbital parameters.):

$$\begin{aligned}m &= 398602 \text{ km}^3/\text{sec}^2 \\ \text{Mean motion} &= n \\ \text{Semi-major axis} &= a \\ n &= (m/a^3)^{1/2} (\text{rad/sec})\end{aligned}$$

$$n_{\text{day}} = n / 2\pi (\text{rev/day}) \text{ units in element set}$$

Since  $m$  is given in  $\text{km}^3/\text{sec}^2$ , we want to convert (rev/day) to (rev/sec):

$$n_{\text{sec}} = n_{\text{day}} (\text{rev/day}) \times (\text{day/sec})$$

$$n_{\text{sec}} = n_{\text{day}} / 86400 \quad (\text{rev/sec})$$

$$n = (2\pi \times n_{\text{day}}) / 86400$$

From the above equation for mean motion:

$$a = (m / n^2)^{1/3}$$

$$a = ((m \times (86400^2)) / (4\pi^2 n_{\text{day}}^2))^{1/3}$$

$$\text{Apogee} = a \times (1 + e) \quad (e = \text{eccentricity})$$

$$\text{Perigee} = a \times (1 - e)$$

The following is the output file header and first few entries for this file:

### SatInfo Build for Space Debris Propagator Model

**Version 1.0**                      **KEO Consultants**

Creates a satellite information file based on the database file  
File Created: Thu Jan 26 10:43:26 1995

Database File: ::#2 Database:Database-1/24/95  
There are 7051 satellite entries in this file

Data Format: >> %05d %15s %15s %8.4f %9.7f %10.4f %11.4f %7.4f<<  
(One leading space, three spaces between parameters)

Start of Satellite Information:

SatNum	Newest Set	Newest Epoch	Incl.	Eccen.	Apogee	Perigee	RCS
00005	twoln-12/19/94	94351.12881707	34.2500	0.1860769	10243.2176	7029.2166	0.3570
00011	twoln-12/16/94	94349.83959258	32.8827	0.1522716	9422.9407	6932.4753	0.7010
00012	twoln-12/19/94	94353.24209383	32.9109	0.1716409	9806.4112	6933.2079	0.7920
00016	twoln-12/19/94	94352.13896339	34.2746	0.2034071	10626.5516	7034.2244	0.6220

The next step in the simulation process was to create a new element-set that realistically represented the distribution of debris in the earth's atmosphere. Various methods were discussed in terms of propagating the elements with time, including random noise fluctuations for the dynamic parameters such as drag, mean motion, and argument of perigee, and averaging of the geometrical parameters such as inclination, right ascension, and eccentricity. It was thought that by creating an idealized set of elements through time using these methods, we might be able to accurately simulate an observation-rate based on this idealized distribution.

Software was written to extract the set of parameters over time for the existing set of real NASA element-sets for particular satellite entries. A few entries with varying RCS values were examined to try and determine the basic behavior of these parameters with time. We expected the geometrical parameters to vary very little over a three month period. But after examining these values, it was found that there was considerable unexpected variation in the behavior of the orbital elements and which it was not obvious how to model mathematically.

An example output of this software for orbital element #00051 is listed below:

### **AnalyzeAv for Space Debris Propagator Model**

**Version 1.0                      KEO Consultants**

Puts the epoch data for an element in Excel format  
File Created: Wed Feb 1 15:08:32 1995

Database File: ::#2 Database:Database-1/24/95

Element analyzed -- Satellite Number: 51

Opened File #0: ::Element Sets:twoln-10/7/94  
Opened File #1: ::Element Sets:twoln-10/10/94  
Opened File #2: ::Element Sets:twoln-10/12/94  
Opened File #3: ::Element Sets:twoln-10/14/94  
Opened File #4: ::Element Sets:twoln-10/17/94  
Opened File #5: ::Element Sets:twoln-10/19/94  
Opened File #6: ::Element Sets:twoln-10/21/94  
Opened File #7: ::Element Sets:twoln-10/24/94  
Opened File #8: ::Element Sets:twoln-10/26/94  
Opened File #9: ::Element Sets:twoln-10/28/94  
Opened File #10: ::Element Sets:twoln-11/1/94  
Opened File #11: ::Element Sets:twoln-11/2/94  
Opened File #12: ::Element Sets:twoln-11/4/94  
Opened File #13: ::Element Sets:twoln-11/7/94  
Opened File #14: ::Element Sets:twoln-11/9/94  
Opened File #15: ::Element Sets:twoln-11/14/94  
Opened File #16: ::Element Sets:twoln-11/16/94  
Opened File #17: ::Element Sets:twoln-11/18/94  
Opened File #18: ::Element Sets:twoln-11/21/94  
Opened File #19: ::Element Sets:twoln-11/23/94  
Opened File #20: ::Element Sets:twoln-11/28/94

Opened File #21: ::Element Sets:twoln-11/30/94  
 Opened File #22: ::Element Sets:twoln-12/2/94  
 Opened File #23: ::Element Sets:twoln-12/5/94  
 Opened File #24: ::Element Sets:twoln-12/9/94  
 Opened File #25: ::Element Sets:twoln-12/12/94  
 Opened File #26: ::Element Sets:twoln-12/14/94  
 Opened File #27: ::Element Sets:twoln-12/16/94  
 Opened File #28: ::Element Sets:twoln-12/19/94

\*\*\*\*\* End of File List \*\*\*\*\*

29 files were opened in the Input List

Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-10/10/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-10/14/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-10/19/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-10/26/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-10/28/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/1/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/2/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/9/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/14/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/18/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/23/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-11/30/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-12/5/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-12/9/94  
 Couldn't find satellite #51 in the Element Set ::Element Sets:twoln-12/16/94  
 Last Epoch for satellite read from set: twoln-12/19/94

Data:

---

SatNum , Epoch , n-dot/2 , n-dot-dot/6 , B\* , Incl. , Ascen. , Eccen. , Arg.Per. , Mean An. , Mean Motion

00051	94280.18633015	-0.00000075	0.00000E 0	0.10000E-3	47.2195	217.1969	0.0105872	341.9418	17.7625	12.18072014
00051	94284.20641116	-0.00000075	0.00000E 0	0.10000E-3	47.2197	204.7649	0.0105946	354.0020	5.9496	12.18072354
00051	94288.88283547	-0.00000075	0.00000E 0	0.10000E-3	47.2191	190.3020	0.0106305	8.1228	352.1242	12.18072437
00051	94293.06700368	-0.00000075	0.00000E 0	0.10000E-3	47.2181	177.3613	0.0106482	20.6654	339.8396	12.18072338
00051	94297.08708019	-0.00000075	0.00000E 0	0.10000E-3	47.2184	164.9298	0.0106684	32.6981	328.0356	12.18072585
00051	94305.94762732	-0.00000076	0.00000E 0	0.10000E-3	47.2183	137.5288	0.0106660	59.1317	301.9913	12.18072974
00051	94310.54196633	-0.00000076	0.00000E 0	0.10000E-3	47.2173	123.3205	0.0106307	72.9260	288.3138	12.18072873
00051	94319.48447009	-0.00000076	0.00000E 0	0.10000E-3	47.2176	95.6651	0.0106155	99.8628	261.4171	12.18073462
00051	94323.50447581	-0.00000075	0.00000E 0	0.10000E-3	47.2171	83.2332	0.0106051	111.9232	249.2859	12.18073398
00051	94331.70854750	-0.00000075	0.00000E 0	0.10000E-3	47.2134	57.8584	0.0105717	136.4871	224.4315	12.18072232
00051	94336.22077389	-0.00000075	0.00000E 0	0.10000E-3	47.2123	43.9029	0.0105544	149.7729	210.9198	12.18071843
00051	94344.17869258	-0.00000074	0.00000E 0	0.10000E-3	47.2093	19.2922	0.0104921	173.5074	186.7086	12.18070444
00051	94347.46030962	-0.00000074	0.00000E 0	0.10000E-3	47.2079	9.1436	0.0104809	183.5463	176.4590	12.18069605
00051	94351.48029124	-0.00000074	0.00000E 0	0.10000E-3	47.2080	356.7125	0.0104489	195.4205	164.3389	12.18069287

Some plotted examples of these parameters follows:

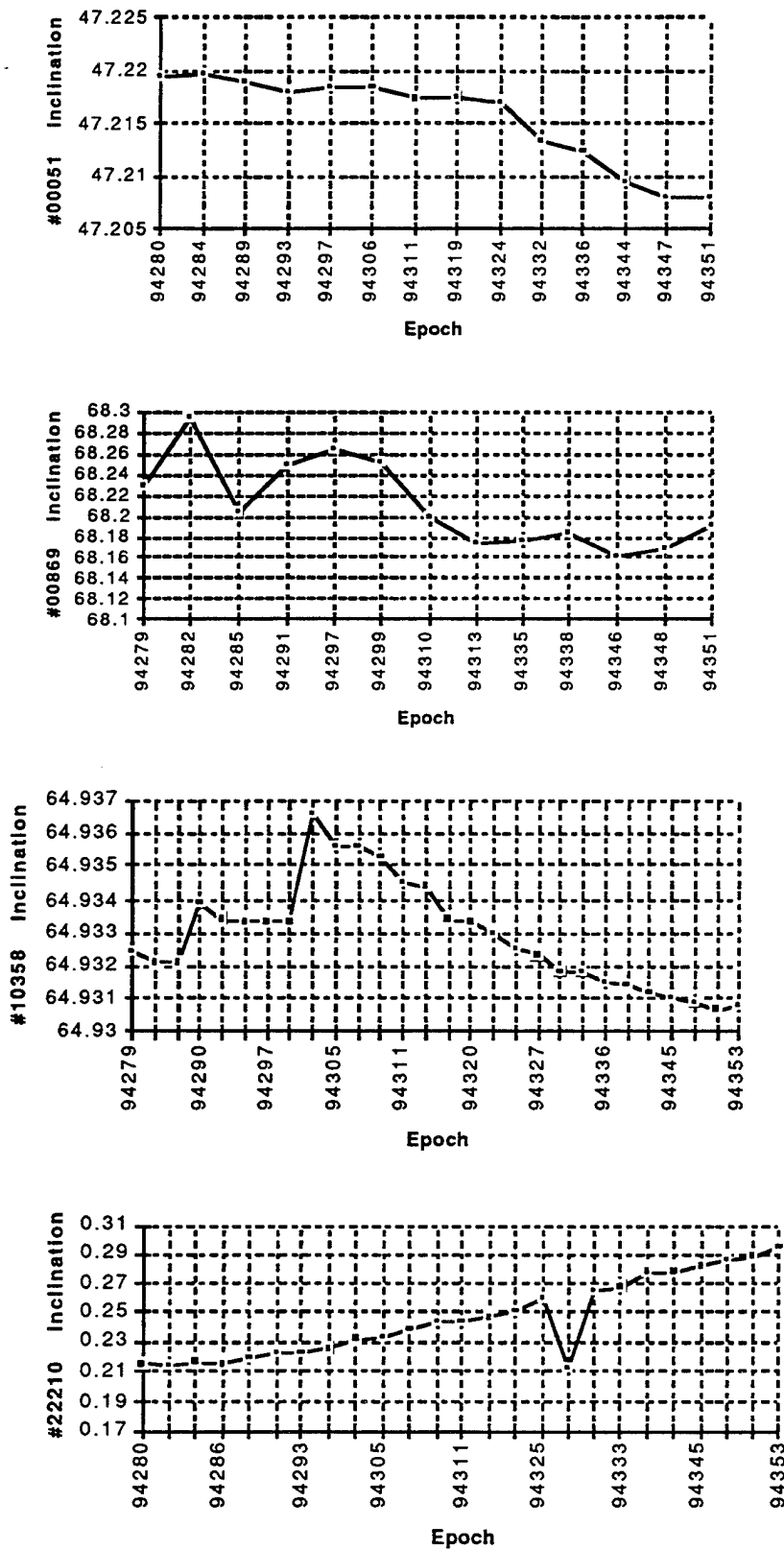


Figure 25: Inclination Plots

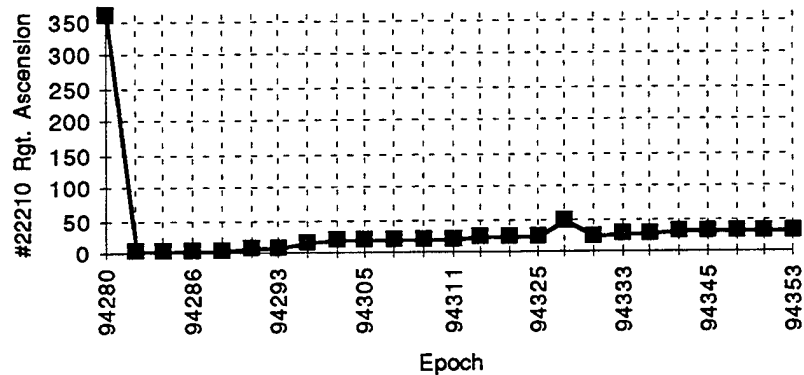
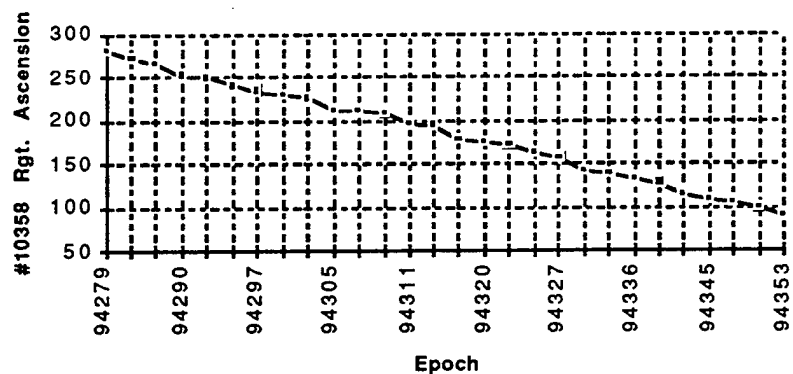
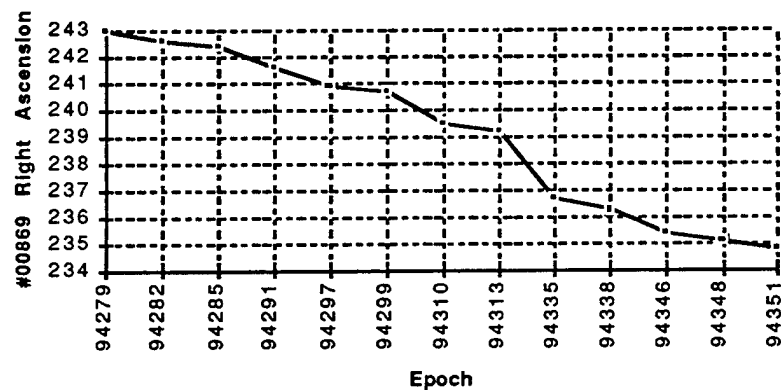
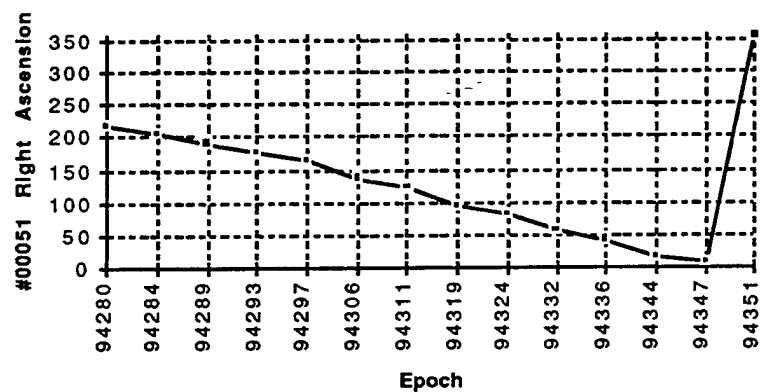


Figure 26: Right Ascension Plots



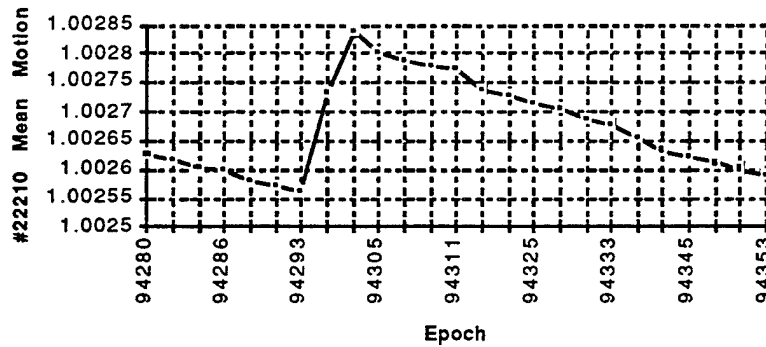
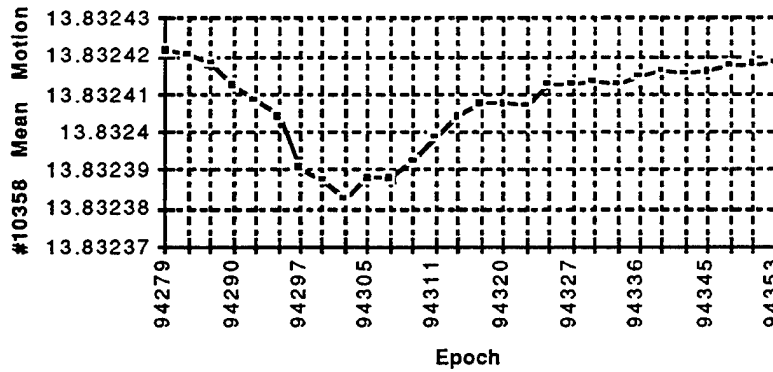
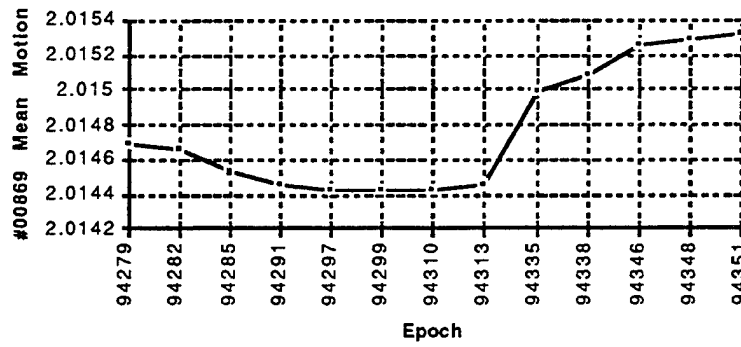
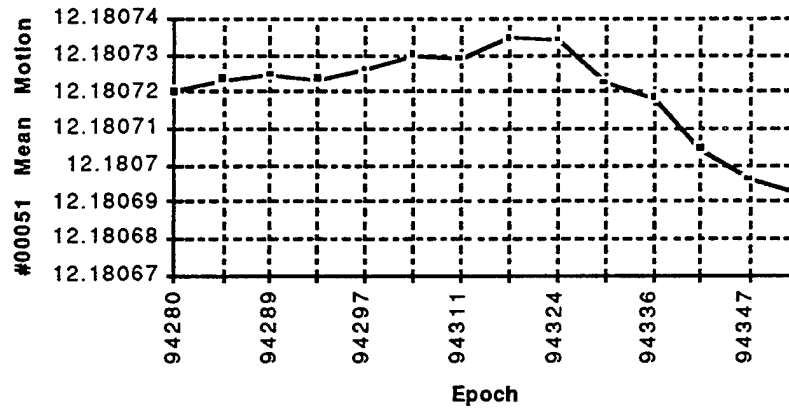


Figure 27: Mean Motion Plots

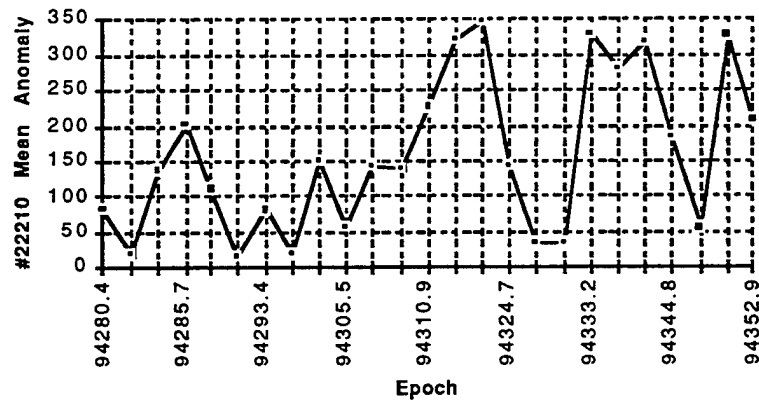
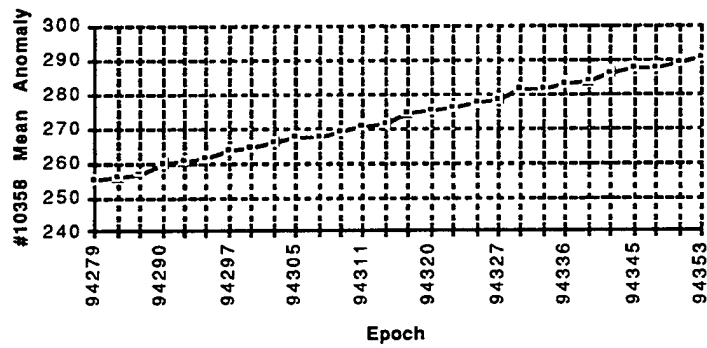
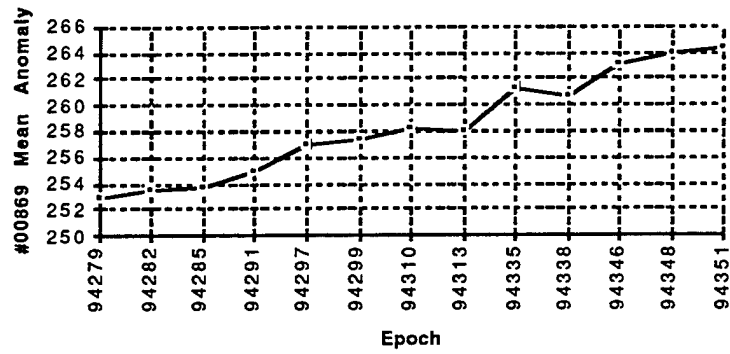
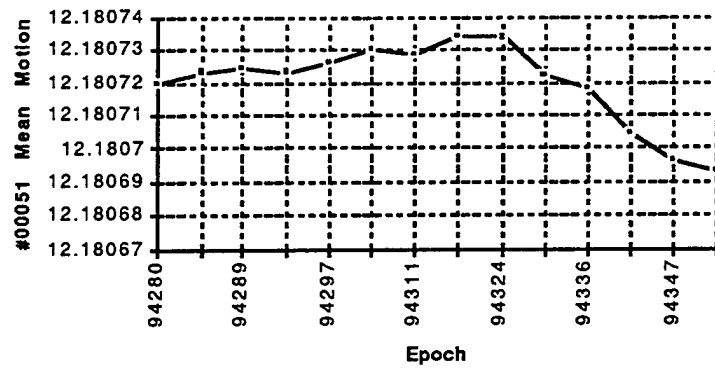


Figure 28: Mean Anomaly Plots

As can be seen by the previous curves, it is difficult to predict the behavior of the orbital parameters. Just a few parameters for a few random objects produces a fairly large spread of curves. To fully understand these curves would be a fairly exhaustive study in itself, and would lead us to needing some kind of curve fitting and extrapolation that would not necessarily lead to any more valid results than a simple averaging.

We decided that our real aim was to produce a distribution of 'idealized' orbits that closely resembled the distribution of valid orbits tracked by NASA. If one 'idealized' element was not detected at a location because its elements were not reflecting the real orbit of the object it represented, there would be another 'idealized' element that would be detected at the same location for the opposite reason. It was thereby assumed that with a data set of 7051 elements, these variations would become insignificant statistically.

In this way we decided to take the time-average of all the elements for the valid objects and to create a new 'Averaged Element Set' based on these results. This 'Averaged Element Set' would then be used by the satellite tracking software ORBITRAK to determine the look-angles (detection rate) for the Rattlesnake Mountain location. This averaged element set was created to have the same format as a NASA two-line element set.

An example output for the Averaged-Element-Set follows:

#### **ComputeAv for Space Debris Propagator Model**

**Version 1.0**

**KEO Consultants**

Creates an Element Set with average values

File Created: Wed Feb 15 15:07:32 1995

Database File: ::#2 Database:Database-1/24/95

Opened File #0: ::Element Sets:twoln-10/7/94  
Opened File #1: ::Element Sets:twoln-10/10/94  
Opened File #2: ::Element Sets:twoln-10/12/94  
Opened File #3: ::Element Sets:twoln-10/14/94  
Opened File #4: ::Element Sets:twoln-10/17/94  
Opened File #5: ::Element Sets:twoln-10/19/94  
Opened File #6: ::Element Sets:twoln-10/21/94  
Opened File #7: ::Element Sets:twoln-10/24/94  
Opened File #8: ::Element Sets:twoln-10/26/94  
Opened File #9: ::Element Sets:twoln-10/28/94  
Opened File #10: ::Element Sets:twoln-11/1/94  
Opened File #11: ::Element Sets:twoln-11/2/94  
Opened File #12: ::Element Sets:twoln-11/4/94  
Opened File #13: ::Element Sets:twoln-11/7/94  
Opened File #14: ::Element Sets:twoln-11/9/94  
Opened File #15: ::Element Sets:twoln-11/14/94  
Opened File #16: ::Element Sets:twoln-11/16/94  
Opened File #17: ::Element Sets:twoln-11/18/94

```

Opened File #18: ::Element Sets:twoln-11/21/94
Opened File #19: ::Element Sets:twoln-11/23/94
Opened File #20: ::Element Sets:twoln-11/28/94
Opened File #21: ::Element Sets:twoln-11/30/94
Opened File #22: ::Element Sets:twoln-12/2/94
Opened File #23: ::Element Sets:twoln-12/5/94
Opened File #24: ::Element Sets:twoln-12/9/94
Opened File #25: ::Element Sets:twoln-12/12/94
Opened File #26: ::Element Sets:twoln-12/14/94
Opened File #27: ::Element Sets:twoln-12/16/94
Opened File #28: ::Element Sets:twoln-12/19/94
***** End of File List *****
29 files were opened in the Input List

```

The following element set contains the averages of 7051 elements

Follows the format of the standard NASA two-line element sets  
Start of Element-Set:

---

```

1 00005U 58002B 94351.12881707 .00000215 00000+0 27520-3 0 9561
2 00005 34.2472 158.5135 1859867 189.0343 168.7877 10.81715520194901
1 00011U 59001A 94349.83959258 .00000202 00000+0 10182-3 0 466
2 00011 32.8770 170.8836 1522124 202.5121 153.7041 11.73957644515012
1 00012U 59001B 94353.24209383 .00000520 00000+0 31197-3 0 3766
2 00012 32.9095 154.2570 1716266 165.4210 195.7307 11.33741238459503
1 00016U 58002A 94352.13896339 .00000069 00000+0 10000-3 0 2756
2 00016 34.2611 134.3475 2034428 163.4567 197.2896 10.46239820481943
1 00020U 59007A 94352.86104870 .00000193 00000+0 79802-4 0 358
2 00020 33.3524 170.5709 1738084 190.8508 169.1716 11.40439000158784

```

This file needs to be modified slightly before being compatible with orbital tracking software such as SATTRAK, TRAKSTAR, and ORBITRAK. To use this element-set, the header should be deleted and one blank line left at the top of the file before the two-line elements start. This element-set then can be read in as a standard NASA two-line element set.

### 7.3 Simulation of Observations:

From the 'idealized' element set described above, we proceeded to calculate predicted observation rates for any given location. We selected Rattlesnake Mountain, WA as a test site for calculations as this was the site from which we performed the ODERACS observations, and it is also likely for future observations. The location of Rattlesnake Mountain is:

```

Latitude:      46.395° North
Longitude:     119.595° West
Altitude:      1066 meters

```

The MacIntosh based software ORBITRAK was used to calculate look angles from the idealized database because of its ease of use and advanced features. Correlations between the results of ORBITRAK and SATRAK (DOS) have been done in association with the ODERACS experiment and found to agree well. The orbital model chosen for the calculations was SGP4/SDP4, and the following is the criterion we used:

Minimum elevation required for detection:	20° from horizon
Time resolution for calculations:	1 minute
Minimum Mean Motion:	1.10
Object illuminated by sun:	At maximum elevation

We did comparison runs with the element-set for time resolutions of 1 minute and 5 minutes and found that, as expected, there were more (about 5%) detections using the higher time resolution. However, the overall shape of the detection curve was unaffected. We chose the 1 minute time resolution for more accuracy and to improve analysis statistics.

The Minimum Mean Motion criterion allows the software to skip over objects with a very small mean motion (very large axes  $> \sim 10,000$  km), and so speeds up the calculations. We did check to see how this affects the statistics for a one month observation period, and found that the number of objects skipped were statistically insignificant ( $< 0.1\%$ ) and in reality most these objects would not be detectable optically.

We ran a one month daily observation histogram and found no significant daily effects, which lead us to choose a 5-day time resolution when analyzing 12 months of data so as to speed up the processing and greatly decrease the size of the Look-Angle output files.

Detection windows were set up to match the optical detection parameters. Both the morning observation and evening observation periods were used. The start/stop criterion for these observations were:

**Civil twilight** (solar depression of  $6^\circ$ ) gives the minimum background emission necessary for optical measurements.

**Shadow Height of 1,000km** (solar depression of  $30.2^\circ$ ) gives the maximum range of objects detectable by ground-based instruments.

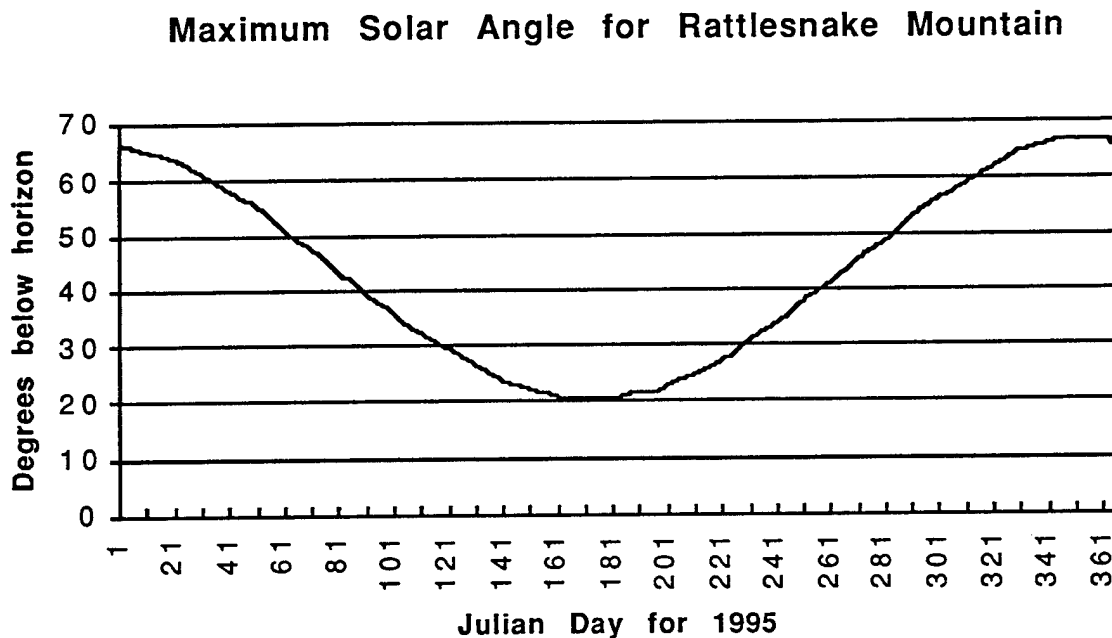
To calculate the solar depression for a shadow height of 1,000km, we used the following calculation:

$$H = R_{\text{earth}} \times (1/\cos\alpha - 1) \quad \text{where } \alpha = \text{solar depression}$$

$$1000 = 6370(1/\cos\alpha - 1),$$

$$\alpha = 30.2^\circ$$

**CalcSunAngle** was written to calculate the start/stop times for both the AM and PM observation windows for Rattlesnake Mountain throughout the year. It was found, however, that as the season moves towards summer, the solar depression never reaches  $30.2^\circ$  as the following curve demonstrates:



**Figure 29: Maximum Solar Angle for Rattlesnake Mountain**

A new criterion was set up for the stop/start times as follows:

AM Observation

**Start:** Solar Depression of 30.19° or when max. depression ends

**Stop:** Solar Depression of 6° (civil twilight)

PM Observation

**Start:** Solar Depression of 6° (civil twilight)

**Stop:** Solar Depression of 30.19° or when max. depression starts

KEO's software calculates these times based on the following calculations:

$$\sin\alpha = -\sin\delta \sin\lambda - \cos\delta \cos\lambda \cos\tau$$

where:  $\alpha$  = solar depression angle

$\delta$  = solar declination

$\lambda$  = latitude

$\tau$  = Hour Angle

From this, we can solve for the Hour Angle:

$$\cos\tau = -(\sin\alpha + \sin\delta \sin\lambda) / (\cos\delta \cos\lambda)$$

To obtain the solar declination, we use the table supplied by Chamberlain, J. W., Physics of the Aurora and Airglow, Academic Press, N.Y., 1961, Chapter 10. **CalcSunAngle** extrapolates the data from this table for the particular date. To calculate the actual local time for the station, we use the following expression:

$$t_{\text{local}} = \tau + 12 - \Delta l - E \quad \text{where } \tau \text{ can be either positive or negative}$$

$\Delta l$  = longitude separation of station from the standard meridian x 4'.

Value is positive if East of meridian, and negative if West of meridian.

E = Equation of time

Take the example of finding Sunrise/Sunset for January 1, 1995 at Rattlesnake Mountain ( $\alpha = 0$ ):

$$\cos\tau = -(\sin(0) + \sin(-23) \sin(46.395)) / (\cos(-23) \cos(46.395))$$

$$= 0.2829 / 0.6349 = 0.4456$$

$$\tau = \pm 63.54^\circ = \pm 4.236 \text{ hours} \quad (15^\circ/\text{hour})$$

From table:  $E = -3 \text{ min.} = -0.05 \text{ hours}$

For Rattlesnake: Longitude =  $119.395^\circ\text{W}$   
Standard Meridian =  $120^\circ\text{W}$  (-8 hours)

$$Dl = 120^\circ\text{W} - 119.393^\circ\text{W} = +.405^\circ = +0.0270 \text{ hours}$$

$$\text{Now, } t = \pm 4.236 + 12 + 0.027 - 0.05$$

$$\text{So } t = 7.787 \text{ hours (sunrise) or } 16.259 \text{ hours (sunset)}$$

These values agree with the ORBITRAK values for sunrise and sunset. For the start/stop windows defined above, the solar depression angles of  $6^\circ$  and  $30.19^\circ$  are used. In cases where the solar depression never reaches  $30.2^\circ$ , the calculation just uses the point where

$$\cos \tau = 1$$

The following curve shows the observation window time for the above calculation:

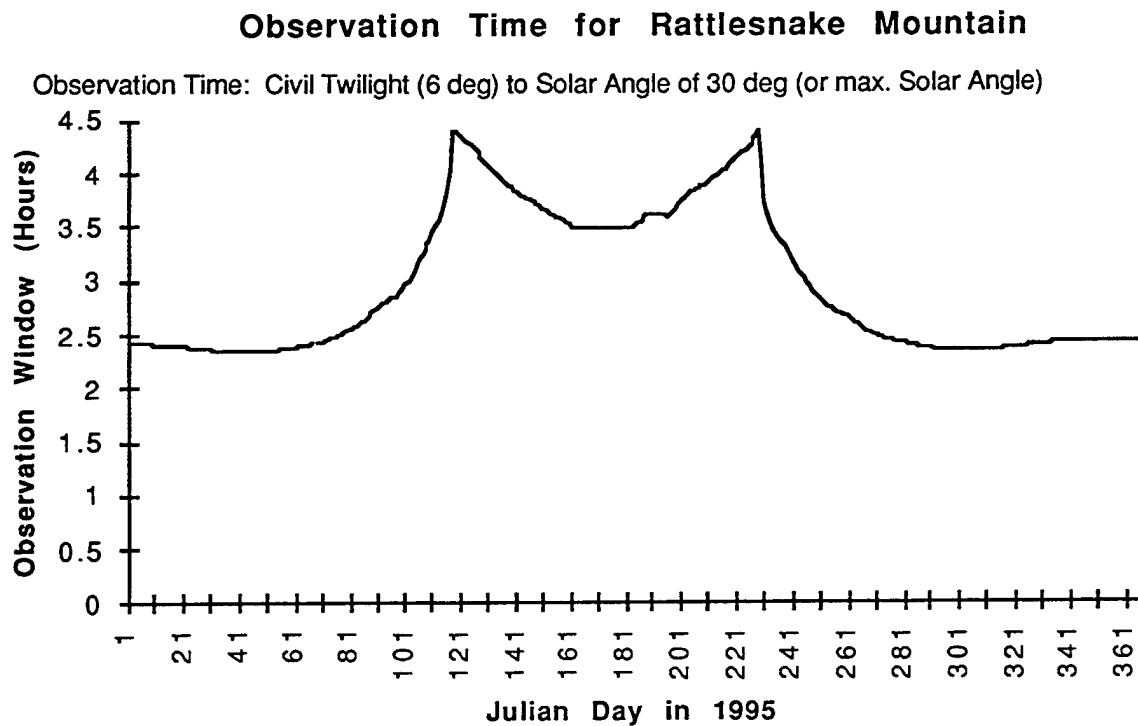


Figure 30: Observation Time for Rattlesnake Mountain



An example of the output file from **CalcSunAngle** is shown below:

Observation times for Solar Depression Angles

For Rattlesnake Mountain Location

Julian Day	Date	Obs. Time	Shadow Hgt. = 1000km	Civil Twilight	Civil Twilight	Shadow Hgt. = 1000km	Max. Solar Angle	Decl.	Eq. of Time
1	1/01/95	2.43987	04:40:16	07:06:39	16:56:05	19:22:29	66.60500	-23.0000	-0.0500
2	1/02/95	2.43746	04:40:08	07:06:23	16:57:13	19:23:28	66.46214	-22.8571	-0.0571
3	1/03/95	2.43508	04:40:01	07:06:07	16:58:20	19:24:26	66.31929	-22.7143	-0.0643
4	1/04/95	2.43274	04:39:54	07:05:52	16:59:27	19:25:25	66.17643	-22.5714	-0.0714
5	1/05/95	2.43044	04:39:46	07:05:36	17:00:34	19:26:24	66.03357	-22.4286	-0.0786
6	1/06/95	2.42818	04:39:39	07:05:20	17:01:41	19:27:23	65.89071	-22.2857	-0.0857

The following curve shows the table values of the Solar Declination (degrees) and the Equation of Time (minutes) that were used for extrapolating these parameters in the above equation.

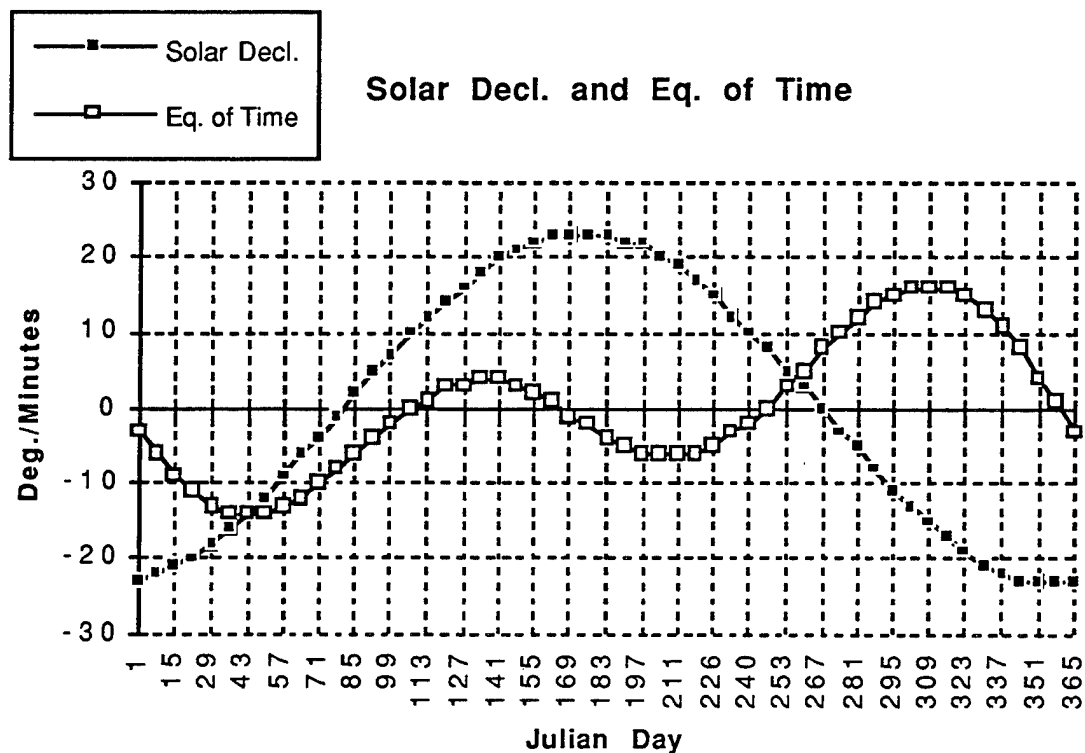


Figure 31: Solar Declination and Equation of Time

## 7.4 Look Angle Calculations:

Using these start/stop times for morning and evening observation windows, the MacIntosh based ORBITRAK software was used to create a list of observed objects for Rattlesnake Mountain. As mentioned above, the criterion used to determine valid observations were:

20° horizon  
1 minute time resolution  
Mean Motion > 1.10  
Object illuminated at the maximum elevation of orbit

An example output ORBITRAK file is shown below:

**OrbiTrack Look - 05/06/95 1854:47**

**Station: Rattlesnake Mountain**

Pass - From: Wed 01/11/95 05:07:15 PM PST  
To: Wed 01/11/95 07:32:18 PM PST

1961 018A

DOW	MM/DD/YY	Time PST HH:MM:SS	Az Deg	El Deg	Range km	Height km	North Lat	West Long	Satellite Visibility	Sun Angle
Wed	01/11/95	6:47:45 PM	151.4	21.0	5644.57	3542.9	16.84	104.18	Illuminated	-22.52 Rise
Wed	01/11/95	7:01:53 PM	74.4	69.2	3682.80	3519.1	47.80	108.69	Illuminated	-24.93 Max
Wed	01/11/95	7:15:35 PM	1.3	21.4	5560.56	3475.5	77.92	116.41	Illuminated	-27.28 Set

1961 018C

DOW	MM/DD/YY	Time PST HH:MM:SS	Az Deg	El Deg	Range km	Height km	North Lat	West Long	Satellite Visibility	Sun Angle
Wed	01/11/95	5:07:33 PM	77.9	25.1	5180.84	3387.1	44.69	78.25	Illuminated	-6.05 Rise
Wed	01/11/95	5:12:37 PM	53.3	29.0	4916.04	3364.8	56.15	80.12	Illuminated	-6.85 Max
Wed	01/11/95	5:20:39 PM	17.0	20.1	5477.15	3335.2	74.36	84.63	Illuminated	-8.11 Set

1961 001

DOW	MM/DD/YY	Time PST HH:MM:SS	Az Deg	El Deg	Range km	Height km	North Lat	West Long	Satellite Visibility	Sun Angle
Wed	01/11/95	6:48:13 PM	152.4	20.3	5663.88	3515.0	16.29	104.56	Illuminated	-22.59 Rise
Wed	01/11/95	7:02:43 PM	72.0	70.1	3700.99	3549.0	48.06	109.18	Illuminated	-25.07 Max
Wed	01/11/95	7:16:46 PM	0.8	20.9	5701.11	3556.7	78.66	117.38	Illuminated	-27.49 Set

As can be seen, ORBITRAK calculates the three orbital parameters for the satellite orbit: Rise, Max and Set. This mode is used to speed up the processing of the data, rather than the normal orbital trajectory mode.

To analyze these output files, we wrote a program called **CreateHist** that goes through all these output files for a given time period and creates an output file in Microsoft Excel format that tabulates the morning, evening and total number of observations for the day of Look-Angle file. **CreateHist** can also select a zenith pointing field-of-view that will discount any objects detected outside of a selected field-of-view. For example, a zenith pointing telescope with a 4 degree field-of-view, would only detect objects who reached a maximum elevation 88° or more (max. of 90°).

Because the observation window time changes with time as shown above, we decided that it would be more appropriate to look at the **Observation Rate** that would be defined as the:

$$(\text{\# of Observations})/(\text{Time of Observation Window})$$

We ran a whole year of observations for 1995 based on the Averaged-Element-Set discussed above for Rattlesnake Mountain and produced the following curves for the following field-of-views (FOV): 140°, 4°, and 1°. The first two curves show the observation histogram and observation rates for the full FOV supplied by the ORBITRAK output files (in this case 140°).

Note that in this analysis, there is no attempt to decide which objects would be actually detectable by an optical equipment at the particular site. The histograms are solely representative of the total number of objects passing through the FOV that are solar illuminated during the observation windows.

### Observation Histogram for Rattlesnake Mountain

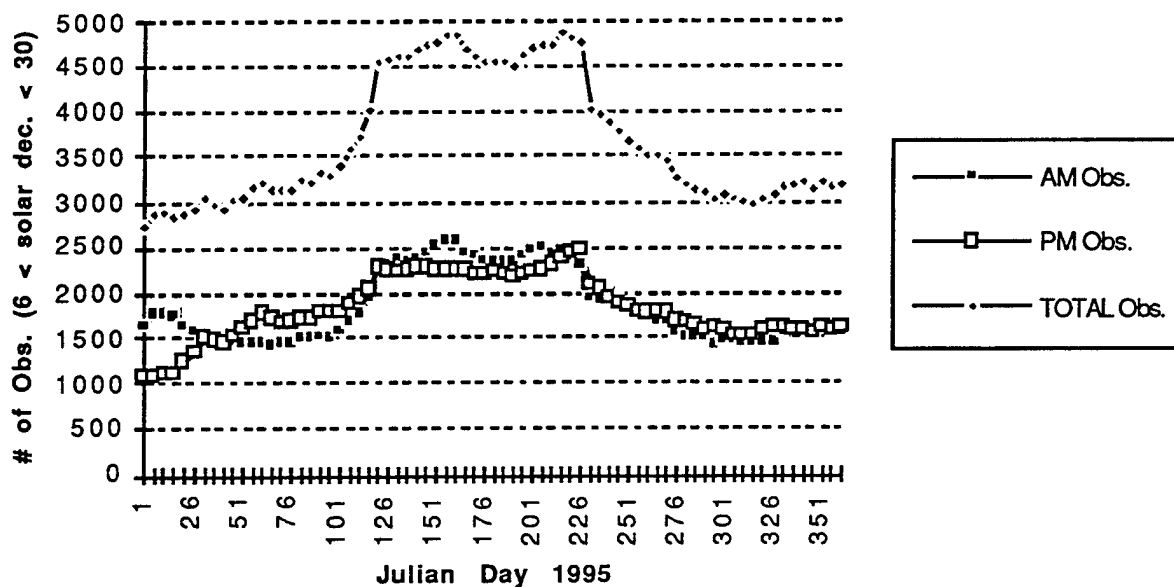


Figure 32a: Observation Histogram for Rattlesnake Mountain

### Observation Rate for Rattlesnake Mountain

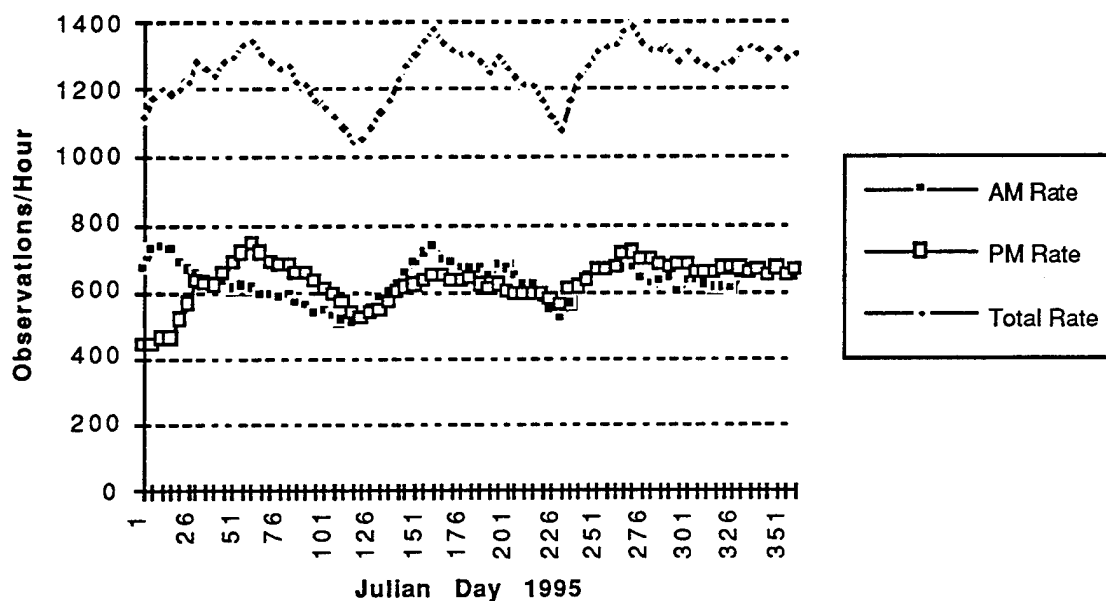
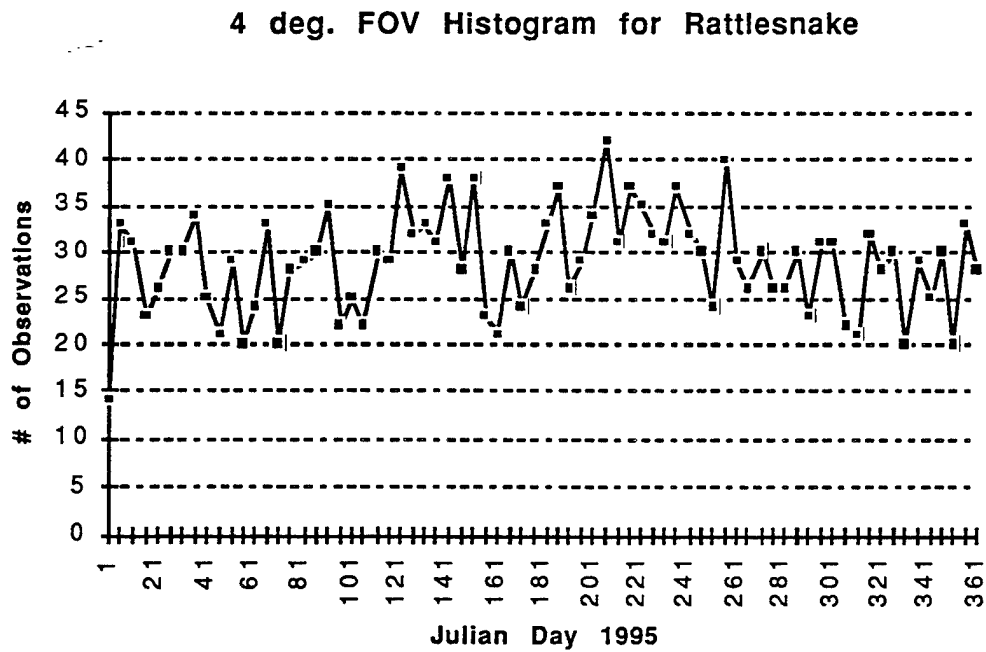
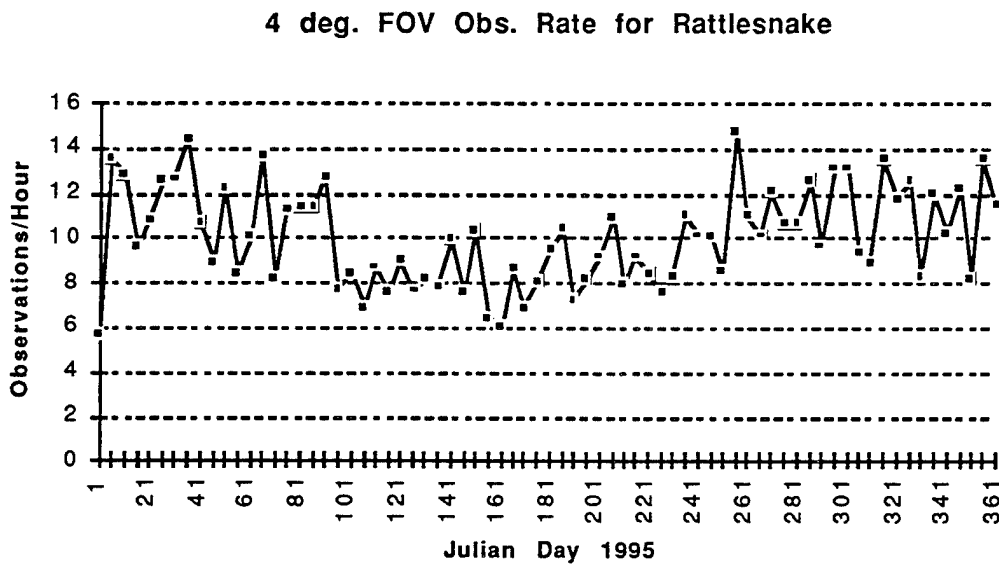


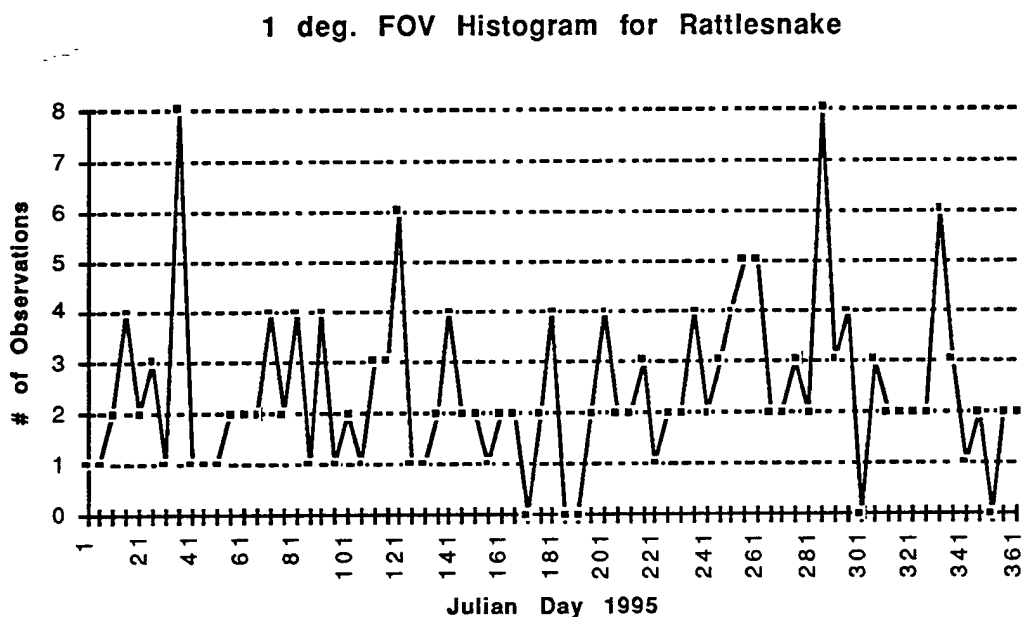
Figure 32b: Observation Rate for Rattlesnake Mountain



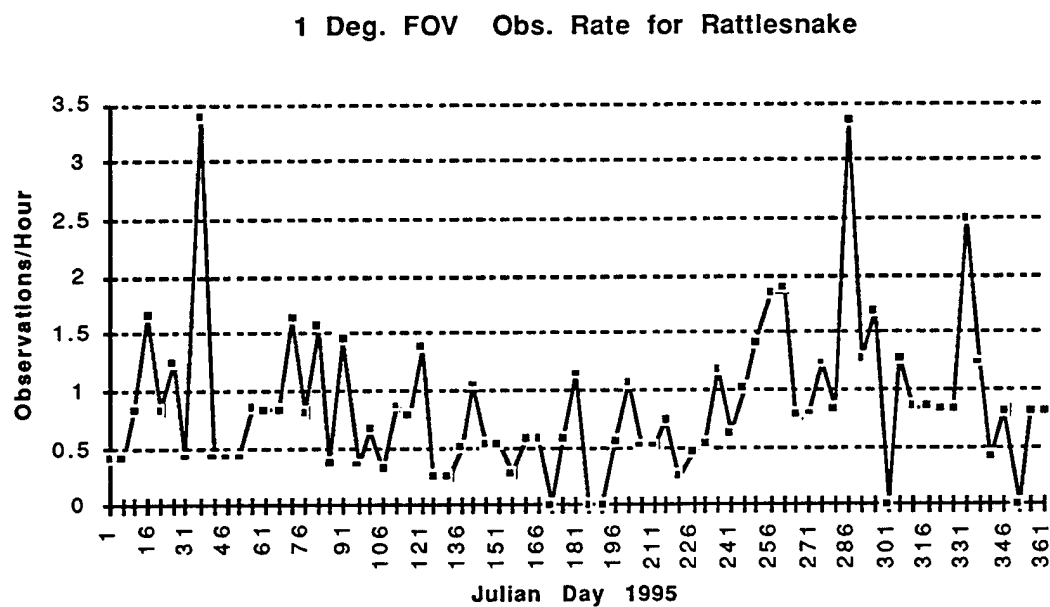
**Figure 33a: 4 Deg. FOV Histogram for Rattlesnake Mountain**



**Figure 33b: 4 Deg. FOV Observation Rate for Rattlesnake Mountain**



**Figure 33c: 1 Deg. FOV Histogram for Rattlesnake Mountain**

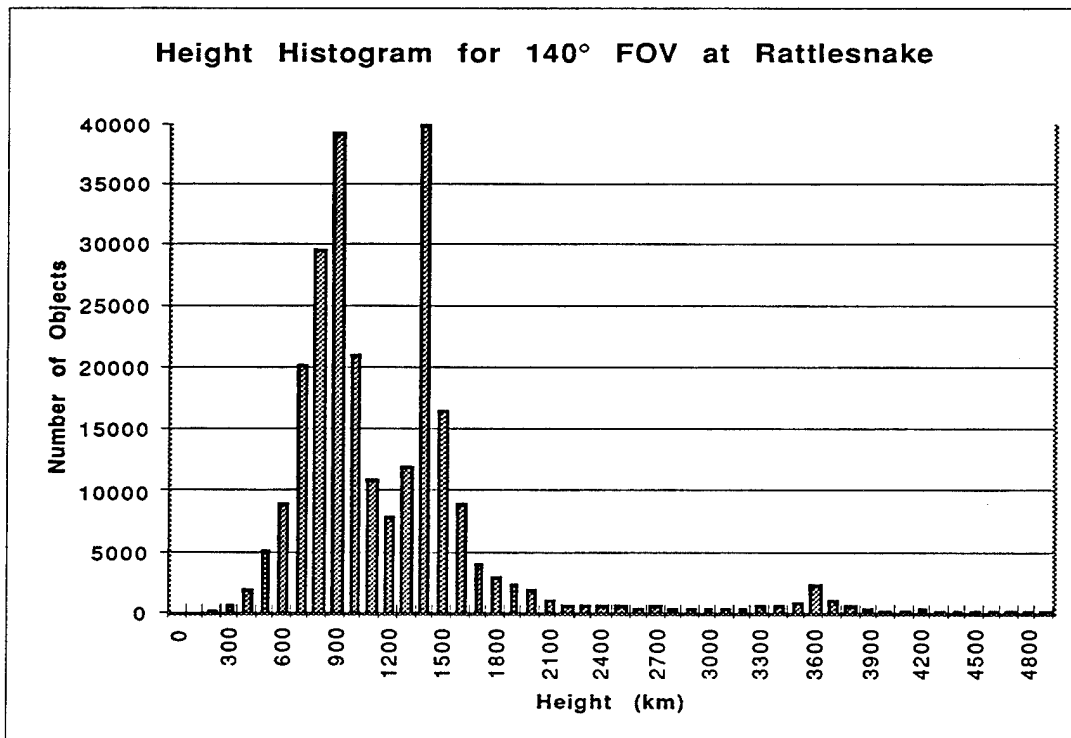


**Figure 33d: 1 Deg. FOV Observation Rate for Rattlesnake Mountain**

In the full FOV curves, where there are enough observations to draw conclusions based on the daily statistics, we can see a seasonal dependence in the observation rate that looks very much like it is inversely related to the observation windows. We can explain these differences by noticing the bias during the summer months towards higher altitude objects. As the sun takes longer to go below the horizon during the summer months and the observation window length increases, we have a longer observation time, but a decreased observation rate due to the diminished distribution of satellites at the higher altitudes. One can note that the minimum of the full FOV observation rate corresponds directly to the longest observation window lengths shown in the Observation Window Time curve.

This reasoning is obviously clouded somewhat by the complicated geometry of satellite distribution effects versus solar geometry and the FOV used for collecting the statistics.

The following curves show certain features of the geometry that need to be considered. The first set of curves shows the height distribution for all objects that were observed during the observation windows used:



**Figure 34a: Height Histogram for 140 Deg. FOV**

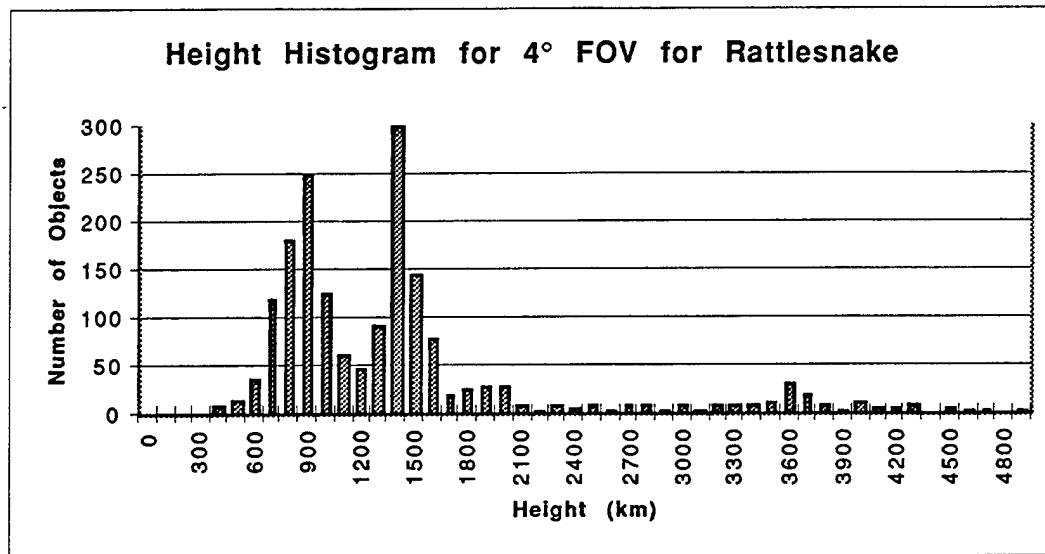


Figure 34b: Height Histogram for 4 Deg. FOV

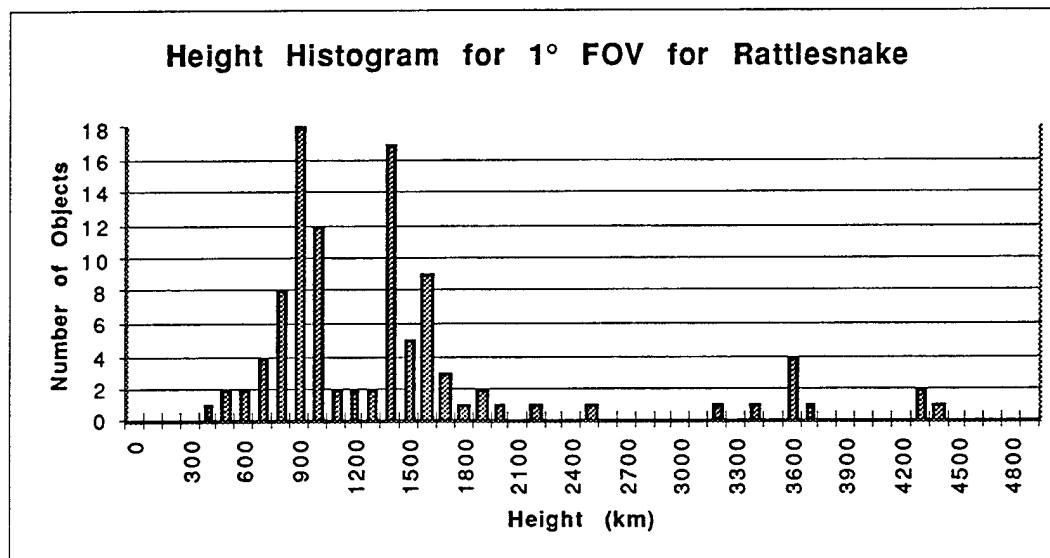


Figure 34c: Height Histogram for 1 Deg. FOV



A few points are immediately evident:

- The height distribution is the same for all three of the Field-of-View restrictions we used.
- The bulk of the objects detected lay in an orbit with a height ranging from about 400km out to about 200 km.
- There is a peak distribution at around 900 km and another peak distribution about 1400 km. There is a smaller peak at around 3600 km.

We can conclude from these curves that the height distribution of these known objects will have a definite effect on the “detection rate” for different observation windows.

If we look at an observation window where the shadow height goes from 35 km ( $6^\circ$ ) to 1000 km ( $30.2^\circ$ ), we can see that the observation rate will change depending on how quickly the sun changes angle. During the winter, the sun sets at a much faster rate than during the peak of the summer.

The next two curves demonstrate the change in solar-depression angle and shadow height over one-half year:

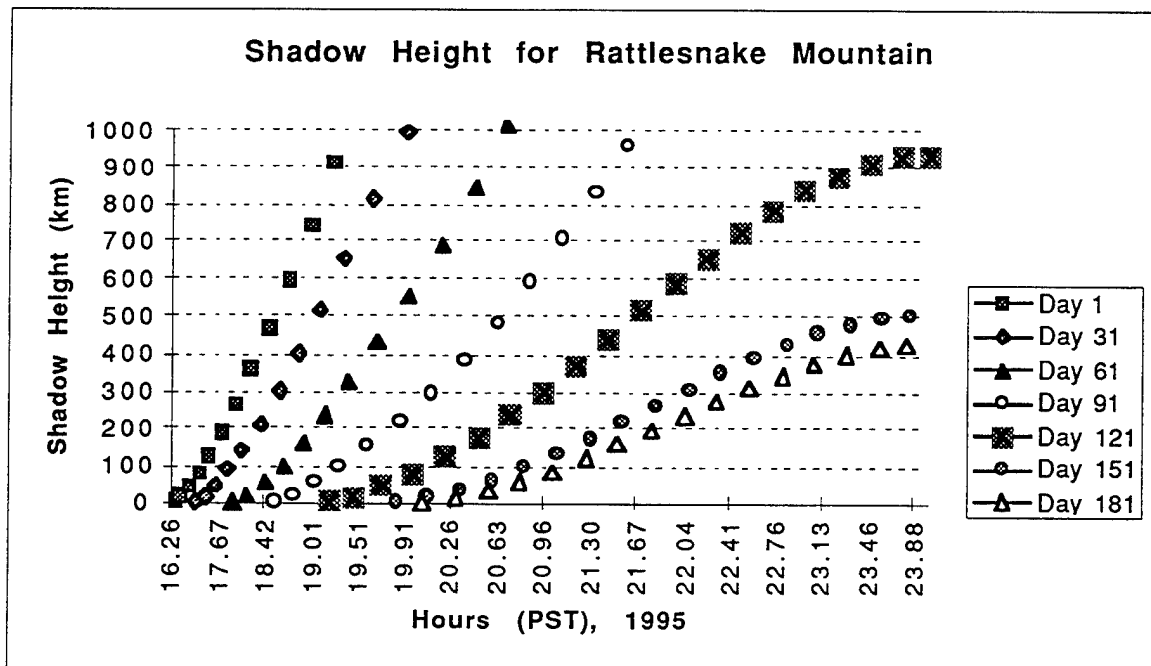


Figure 35a: Shadow Height for Rattlesnake Mountain

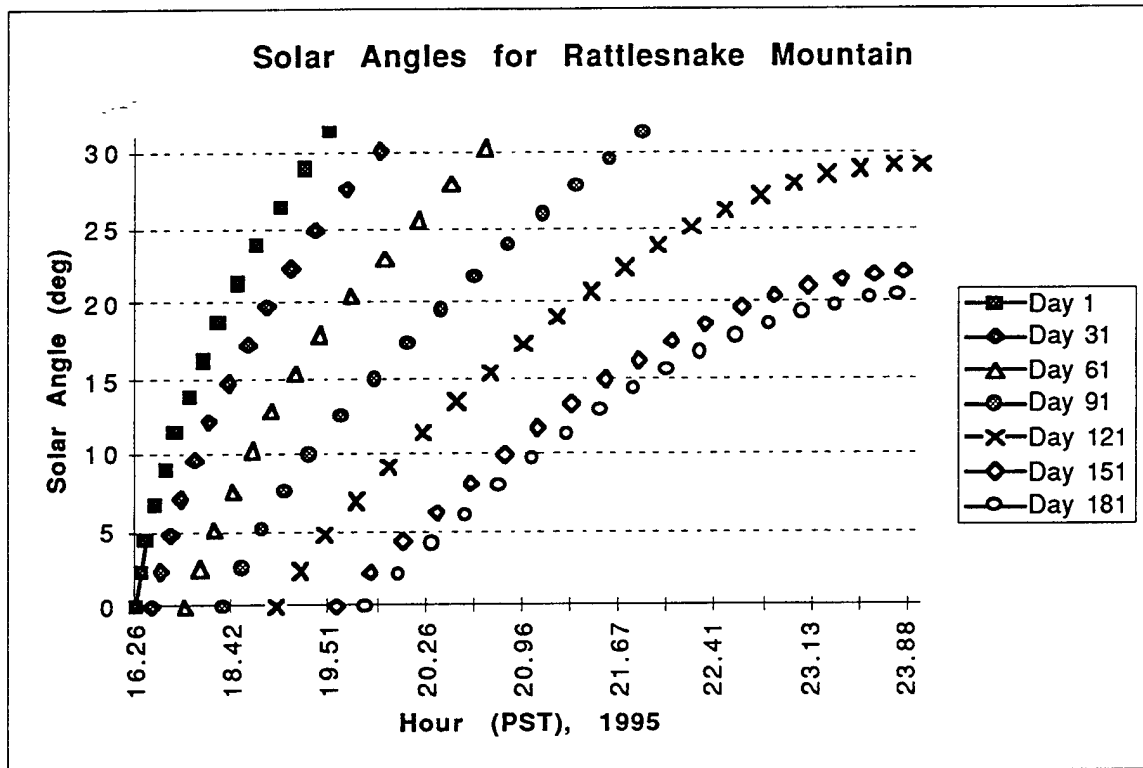


Figure 35b: Solar Angles for Rattlesnake Mountain

As can be seen from the above curves, there is a great difference between the observation times for objects between a height of, say, 400 to 1000 km during the beginning of the year, than there is as we reach the summer solstice.

## 7.5 Conclusions:

These seasonal variations make it difficult to predict an average daily observation rate for any geographical location using this method. It would seem that to accurately determine a predicted "detection-rate" for a given location, we would need to do yearly studies to build up statistics on the seasonal fluctuations. This would have additional problems due to the changing nature of object distribution with time over periods of one year or longer.

Conversely, it would be difficult to use short-term (~ months) debris observations from a particular location to derive conclusions about any changes in the debris population, as such changes would be expected to be small compared to seasonal effects. Probably only yearly averages would be a good indicator of real changes in the overall debris population.

## 8. References:

1. Chamberlain, J. W., "Physics of the Aurora and Airglow", Academic Press, N.Y., 1961.
2. Chu, P.L., "Efficient Detection of Small Moving Objects, MIT Lincoln Laboratory Technical Report No. 846, 1989.
3. P. Dao, "Orbital Debris Environment: an Update", AIAA-94-592, 32nd Aerospace Sciences Meeting, Reno, NV, January 1994.
4. P. Dao, A. Wilson and A. Reinhardt, "Space Debris Optical Measurement: Use of Stare Sensors", AIAA-93-159, 31st Aerospace Sciences Meeting, Reno, NV, January 1993
5. Eather, R. H., "Space Debris Detection", Phillips Laboratory Report # PL-TR-93-2058, Dec. 23, 1993, ADA269254
6. Eather, R.H. and Vu, Q., Automatic Real-time Debris Detection System (ARDD), Keo Consultants and VuSystems, Inc., 1994
7. Henize, K.G., J.F. Stanley, C.A. O'Neill, and B.S. Nowakowski, in "Space Debris Detection and Mitigation", SPIE Proceedings, Vol. 1951, 76-85, 1993.
8. Hussain, Z. "Digital Image Processing, Practical Applications of Parallel Processing Techniques". Ellis Horwood, 1991.
9. Pohlig, S.G., "Maximum Likelihood Detection of Electro-optic Moving Targets", MIT Lincoln Laboratory, Technical Report 940, 16 January 1992.
10. Sveadlow, M. "Hough Transform Debris Automated Detection", AIAA-93-162, 31st Aerospace Sciences Meeting, Reno, NV, January 1993.
11. Teledyne Brown Engineering, Sample Catalog of Small Objects. 5th Ed., Colorado Springs, CO, Feb. 1986

### Technical References for DSP Processing:

1. Interagency Group Rept. on Orbital Debris for National Security Council, Feb. 1989
2. D. King, "Using DSP Chips", VMEbus Systems, Vol. 11, No. 2, April 1994.
3. MegaImager Specifications, Applied Silicon Inc., 1994.
4. MegaImager Hardware User's Guide, Applied Silicon Inc., 1994.
5. MegaImager Software User's Guide, Applied Silicon Inc., 1994.
6. P360F Power Grabber Specifications, Dipix Technologies Inc., 1993.
7. TMS320C4x User's Guide, Texas Instruments, Inc., Literature Number SPRU063, 1991.
8. TMS320C4x Technical Brief, Texas Instruments, Inc., 1991.
9. TMS320 Floating-Point DSP Optimizing C Compiler, Texas Instruments, Inc., 1993.
10. TMS320C80 Multimedia Video Processor (MVP) Technical Brief, Texas Instruments, Inc., 1994.
11. Tiger 40/PC Specifications, DSP Research, 1994.
12. Tiger 440 Specifications, DSP Research, 1994.
13. TMS320C80 Technical Brief, Texas Instruments, Inc., 1994.